DOCUMENT RESUME

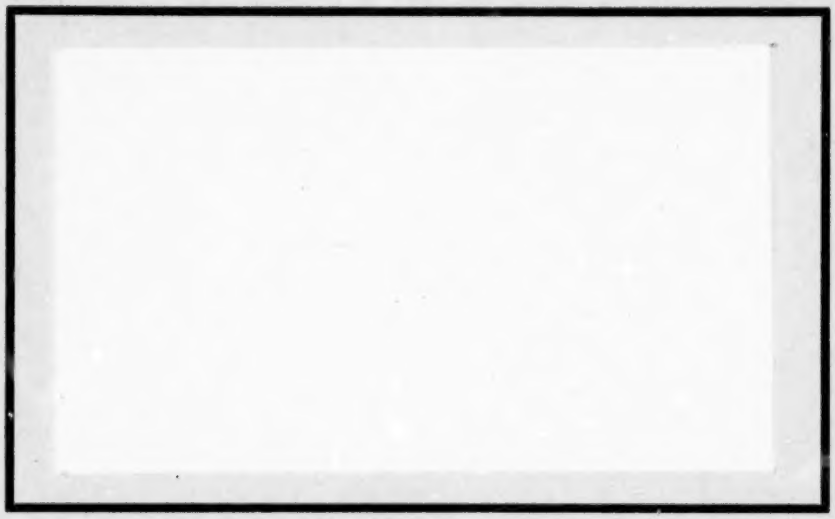
ED 084 876 EM 011 700

AUTHOR	Smith, Authella; And Others
TITLE	A Coursewriter II Function (FCALC) For the Manipulation of Numerical and Algebraic Expressions. Systems Memo Number One.
INSTITUTION	Texas Univ., Austin. Computer-Assisted Instruction Lab.
SPONS AGENCY	Department of Defense, Washington, D.C. Advanced Research Projects Agency.; National Science Foundation, Washington, D.C.
REPORT NO	SM-1
PUB DATE	Oct 69
NOTE	89p.; See also EM 011 701 and EM 011 702
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Assisted Instruction; *Computer Programs; *Computer Science; Logic; *Programing; Technical Reports
IDENTIFIERS	Algebraic Expressions; *Coursewriter II Function FCALC; IBM 1500 Instructional System; Numeric Expressions; Scientific Notation

ABSTRACT

Documentation of the Coursewriter II Function FCALC is provided. The function is designed for use on the IBM 1500 instructional system and has three major applications: 1) comparison of a numeric expression in buffer 5 with a numeric expression in buffer 0; 2) comparison of an algebraic expression in buffer 5 with an algebraic expression in buffer 0; and 3) evaluation of a numeric expression in buffer 0 with return of a numeric value to buffer 0 in scientific notation display cards. The report includes a user's guide which discusses usage rules and the application of the function, an installation guide to assist programers in installing the function, and a program logic section which outlines the purposes of the function's routines and subroutines. Program modification notes and an appendix containing a source listing of the entire function are also included. (Author/PB)

ED 084876



THE UNIVERSITY OF TEXAS AT AUSTIN

Computer Assisted Instruction Laboratory

AUSTIN

EM 011 100

FILMED FROM BEST AVAILABLE COPY

ED 084876

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

A COURSEWRITER II FUNCTION (FCALC) FOR THE MANIPULATION OF NUMERICAL AND ALGEBRAIC EXPRESSIONS

Systems Memo No. 1

October, 1969

Authella Smith
Matthew Wheaton
Carl Gregory
C. Victor Bunderson

Co-Sponsored By

The National Science Foundation
Grant GJ 509 X

and

Advanced Research Projects Agency
ARPA Order No. 1261
Contract N00014-67-A-0126-0006

Reproduction in whole or in part is permitted for any purpose of the United States Government.

This document has been approved for public release and sale; its distribution is unlimited.

Computer-Assisted Instruction Laboratory
The University of Texas at Austin
Austin, Texas 78712

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Advanced Research Projects Agency
Washington, D. C. 20301

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
A COURSEWRITER II FUNCTION (FCALC) FOR THE MANIPULATION OF NUMERICAL AND ALGEBRAIC EXPRESSIONS

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*
Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*
A. Smith, M. Wheaton, C. Gregory, & C. V. Bunderson

6. REPORT DATE
October 1969

7a. TOTAL NO. OF PAGES
88

7b. NO. OF REFS
0

8a. CONTRACT OR GRANT NO.
N00014-67-A-0126-0006

b. PROJECT NO.
NR 154-282/8-13-68 Code 458

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
N1-69

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
U. S. Department of the Navy
Office of Naval Research
Washington, D. C.

13. ABSTRACT

The Coursewriter II Function FCALC was designed for use on the IBM 1500 Instructional System. This function has three major applications: (1) Comparison of a numeric expression in buffer 5 with a numeric expression in buffer 0; (2) Comparison of an algebraic expression in buffer 5 with an algebraic expression in buffer 0; and (3) Evaluation of a numeric expression in buffer 0 with return of a numeric value to buffer 0 in scientific notation, display cards. This report provides documentation of the Function FCALC. (U)

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Coursewriter II Function						
Numerical & Algebraic Expressions						
Buffer						
Scientific Notation						
Display Code						
Numeric Comparison						
User's Guide						
Installation Guide						
Program Logic						
Program Modification						

PREFACE

This technical report provides documentation of the Coursewriter II Function FCALC for use on the IBM 1500 Instructional System It is comprised of five parts:

- Function FCALC Users Guide in which the application of the function and the usage rules are discussed from a users standpoint.
- Function FCALC Installation Guide which contains the procedures to be followed by the systems programmers when installing the function.
- Function FCALC Program Logic in which is outlined the purposes of the various routines and subroutines which comprise the function.
- Program Modification Notes which contain procedures and suggestions to follow when modifying the function.
- Appendix A which contains a source listing of the entire FCALC function.

The Coursewriter II function was developed through funds provided by National Science Foundation Grant GJ 509 X. The needs for and the requirements of the function were specified by Mrs. Authella M. Smith, instructional designer and author, and Mr. Carl Gregory, CAI programmer. The system analysis and programming was done by Mr. Matthew Wheaton of Booz-Allen Applied Research, Incorporated.

TABLE OF CONTENTS

		Page
I.	FUNCTION FCALC USERS GUIDE	1-1
1.	Function Application	1-1
2.	Editing Rules	1-3
3.	Syntax and Evaluation Rules	1-4
4.	Calling Parameters	1-6
5.	Error Returns	1-7
II.	FUNCTION FCALC INSTALLATION GUIDE	2-1
1.	Initial Installation	2-1
2.	Function testing	2-3
III.	FUNCTION FCALC PROGRAM LOGIC	3-1
1.	The Absolute Text Loader	3-1
2.	The interpreter/Evaluator	3-2
3.	The Output Converter	3-2
4.	Operational Subroutines	3-3
5.	Floating Point Subroutines	3-16
IV.	PROGRAM MODIFICATION NOTES	4-1
1.	Label Relocation	4-1
2.	Relative Error	4-1
3.	New Algorithm Addition	4-2

APPENDIX A - Source Listing for Coursewriter II Function FCALC

I. FUNCTION FCALC USERS GUIDE

I. FUNCTION FCALC USERS GUIDE

1. FUNCTION APPLICATION

Coursewriter II function FCALC has three major applications: (1) comparison of a numeric expression in buffer 5 with a numeric expression in buffer 0; (2) comparison of an algebraic expression in buffer 5 with an algebraic expression in buffer 0; and (3) evaluation of a numeric expression in buffer 0 with return of a numeric value to buffer 0 in scientific notation, display code.

(1) Numeric Comparison

The expression found in buffer 0 is evaluated first. If any syntax or arithmetic error is found, transfer is made to the error routine and no further analysis is made. If no errors are found, the resulting numeric value is stored. The expression found in buffer 5 is evaluated next. As before, on any error, transfer is made to the error routine. If no error is found, the resulting value of this expression is compared with the resulting value from buffer 0 according to the following formula:

$$|S - A| \leq |FA|$$

where S is the value of the expression in buffer 5, A is the value of the expression in buffer 0, and F is the acceptable relative error. At present F has been given a value of 0.01. The installation systems programmer may easily change it to any desired value. If the formula is satisfied, i.e., $|S - A|$ is less than or equal to $|FA|$, the two expressions are considered equal, and the contents of counter 22 and counter 23 are replaced by zero. Otherwise, the two expressions are considered unequal, and the contents of counter 22 and counter 23 are replaced by -1. No change is made to the contents of buffer 5 or buffer 0. Control is then returned to the user.

(2) Algebraic Comparison

Every Coursewriter II character other than +,-,x, ÷,/,(,),., 1,o, and the digits 0 through 9 will be considered an algebraic symbol. Each symbol will be given a numeric value. These values are calculated by taking the low order four bits of the symbol's EBCDIC representation and adding the numeric value two. This converts an algebraic expression into a numeric expression. Comparison is then made of the expressions in buffer 0 and buffer 5 as if they were numeric. Again, counter 22 and counter 23 are used to hold comparison codes. The contents of buffers 5 and 0 are not modified by the algebraic replacement.

(3) Numeric Evaluation

The expression in buffer 0 is evaluated exactly as in numeric comparison. The resulting value is then converted into scientific notation and placed in buffer 0. The characters in buffer 0 will be in a form suitable for display with the DT operation code. Buffer 5 will not be analyzed, or changed. The contents of buffer 0 will be replaced by the resulting value. The contents of counter 22 and counter 23 will be modified only in case of error.

2. EDITING RULES

Certain characters and character sequences receive special treatment by the editing routines. Generally, this is to ease format restrictions for the user and to compensate for certain common input mistakes. Backspaces, carriage returns, and blank characters are completely ignored. The equal sign is treated as an EOB. This means that if a user enters an equation rather than an expression, only the expression to the left of the equal sign will be evaluated. Pairs of index, reverse index, or reverse index, index will be ignored. If an index character is found immediately following the characters "LOG", it and all succeeding characters up to and including the next reverse index are ignored. This permits the user to write in LOG_{10} (...., if desired. The character "/" will be treated as if it were " $\frac{a}{c}$ ". The letter "x" will be treated as a times sign for numeric comparison or numeric evaluation. For algebraic comparison,

it will be treated as the symbol "x". Upper and lower case alphabetic characters will be considered as identical. The characters "l" and "o" will be treated as a numeric "1" and "0" respectively, unless they are found in the sequence "log", in which case, the sequence will be considered as an instruction to interpret the following argument according to the logarithm base ten function. The characters +,-,x, $\frac{0}{0}$,(,) , . , ↑ , (reverse index), ↓ (index),0,1,2,3,4,5,6,7,8, and 9 will be handled according to their mathematical significance. All other characters will be considered invalid in numerical comparison or evaluation. In algebraic comparison, they will be considered algebraic symbols.

3. SYNTAX AND EVALUATION RULES

Evaluation of expressions proceeds from left to right according to the following operation hierarchy.

- Parenthesized expressions
- Logarithm function
- Power expressions
- Multiplications and divisions
- Additions and subtractions

Note that this conforms to neither FORTRAN nor APL specifications, but is closer to standard mathematical notation. Hence, it should pose no learning problems.

Both implied and explicit multiplication is permitted. The following forms of implied multiplication are valid.

- (1)(2) is treated as 1X2
- 3(4) is treated as 3X4
- (5)6 is treated as 5X6
- $7^{8}9$ is treated as $7^{8}X9$
- (10)log(11) is treated as 10Xlog(11)
- 12 log(13) is treated as 12Xlog(13)
- 14^{15}log(16) is treated as 14^{15}Xlog (16)
- ab is treated as aXb (algebraic comparison only)
- 17c is treated as 17Xc (algebraic comparison only)
- d18 is treated as 16Xd (algebraic comparison only)

When a division operation is found in a string of multiplication operations, division is done only with the immediately succeeding operand. Thus, 1X2 ÷ 3X4 is treated as $\frac{1X2X4}{3}$, not as $\frac{1X2}{3X4}$. If the latter is desired, the expression should be written 1X2 ÷ (3X4).

The reverse index is used to denote a power expression. This moves the cursor up one line. The entire expression written on this line will be considered the power by which to raise the base. As many levels of power expression as desired are permitted. The expression $2^{3^{2^{1}}}$ is evaluated as:

$2^{3^{2^{1}}} = 2^{3^{1}} = 2^{1} = 2$, not as $((2^{1})^{2})^{3} = 2^{6} = 64$. The computation algorithm for power expressions is found in Exhibit I-1.

The characters "log" denote a call to the logarithm base 10 function. The argument to this call must be placed in parenthesis. An expression such as log $(10)^2$ will be treated as $(\log(10))^2 = 1$, not as log $((10)^2)=2$. Parenthesized expressions may be nested to any level if they do not contain the log function. Thus, (((((2)(3))(4))(5))) is a valid expression which is treated as 2X3X4X5. However, if nested expressions contain the log function, they may be nested no more than four deep. The expression $100+\log(10^{10}\log(10^{10}\log(10^{10}\log(10^{10}))))$ shows the maximum level of permissible nesting.

Numeric operands may contain up to four significant digits. Any number of zero digits may precede or follow the significant digits. Thus the number 157600000000000000000 is acceptable but 156000000000000000001 is not. A number may be written with or without a decimal point, as desired. The range of permissible values for numeric input, output, or intermediate result is approximately $10^{-38} \leq |N| \leq 10^{38}$ or 0.

4. CALLING PARAMETERS

The Coursewriter II function FCALC is accessed by using the FN operation code as follows:

FN FCALC/" where D is a numeric digit 0 through 4. If the parameter is:

- 0, numeric comparison is made. A request to raise a negative base to a non-zero power will result in an error return.

- 1, numeric comparison is made. A request to raise a negative base to a non-zero power will be evaluated according to Exhibit I-1.

- 2, algebraic comparison is made. Powers are treated as if the parameter were 1.

- 3, numeric evaluation is made. Powers are treated as if the parameter were 0.

- 4, numeric evaluation is made. Powers are treated as if the parameter were 1.

5. ERROR RETURNS

The error routine places the error number in counter 23. The character number $\pm$ 1 of the input expression where the error was discovered is placed in counter 22. In the case of computation errors 1, 2, 3, 4, 5, 6, or 21, counter 22 will always have the number of the last character in the expression, since the entire expression is scanned before computation begins. The list of error codes is found in Exhibit I-2.

EXHIBIT I-1
a^b Computation
Algorithm

When raising a^b, the following algorithm is used:

- $a > o$, $b < o$ -- return $e^{b \times \ln(a)}$
- $a > o$, $b = o$ -- return 1
- $a > o$, $b > o$ -- return $e^{b \times \ln(a)}$
- $a = o$, $b < o$ -- return error code 4
- $a = o$, $b = o$ -- return error code 5
- $a = o$, $b > o$ -- return 0
- $a < o$, $b < o$ -- if calling parameter is 0 or 3, return error code 6. Otherwise, compute $c = \frac{1}{e^{-b \times \ln(-a)}}$. Correct b to nearest integer. If odd, return -c. If even, return c.
- $a < o$, $b = o$ -- return 1
- $a < o$, $b > o$ -- if calling parameter is 0 or 3, return error code 6. Otherwise, compute $c = e^{b \times \ln(-a)}$. Correct b to nearest integer. If odd, return -c. If even, return c.

EXHIBIT I-2
Error Return Codes

1. Floating Overflow $|N| > 1.7 \times 10^{38}$.
2. Floating Underflow $0 < |N| < 10^{-38}$.
3. Attempted Division by Zero.
4. Attempted to Raise Zero to a Negative Power (0^{-P}).
5. Attempted to Raise Zero to the Zero Power (0^{0}).
6. Attempted to Raise a Negative Number to a Non-Integer Power.
7. Two Operators in Succession.
8. Expression Begins with DIADIC Operator.
9. Non-numeric Operand or Illegal Function Name.
10. Expression Ends with Operator.
11. Null Expression, or Expression Begins with ")".
12. Mismatched Parenthesis or Index Error.
13. Number Contains More Than One Decimal Point.
14. Number Contains More Than Four Significant Figures.
15. Log Function Call Not Followed by "(".
16. "(" Followed by Reverse Index, or Null Power Expression.
17. Power Expression Ends with Operator.
18. Null Response, or "(" Only, or Expression Ends with Reverse Index.
19. Operand Table Overflow.
20. Over Four Nested Functions.
21. Attempted to Take Log of Zero or Negative Number.
22. Numeric Operand Contains "." but No Digits.

II. FUNCTION FCALC INSTALLATION GUIDE

II. FUNCTION FCALC INSTALLATION GUIDE

1. INITIAL INSTALLATION

Coursewriter II function FCALC is divided into two parts. One part, the relocatable part, is loaded by the interpreter into one of the function areas. This part then serves as a loader to bring the other portion, the absolute portion, into the application two area. Loading of the two parts must be done in the following sequence:

(1) System Pack Allocation

An area of six contiguous sectors on the system pack which are not being used must be found. Preferably, this should be in the fixed area. However, if no space is available there, a dummy course may be registered on the system pack, and its reserved area used. If a dummy course is registered on the system pack, the systems programmer should change the first character of the name in the system directory to a blank. This will prevent inadvertent deletion of the course, and hence the fixed portion of the function.

(2) Card-To-Disk Preparation

An ABC card for the card-to-disk program must be made in the following format:

```
ABC CF000 6802 NNN 6  *FCALC BY BAARINC.
```

NNN is to be replaced by the first sector numer of the 6 contiguous free sectors.

(3) Reassembly

Portion 1, the relocatable section, must be reassembled. The statement:

```
SECAD   EQU   270
```

should be replaced by:

```
SECAD   EQU   NNN
```

where NNN is the same as on the ABC card.

(4) System Loading

Using the card-to-disk program, load the absolute binary deck onto the Master pack. Use the IBM supplied reconfiguration program to reconfigure the system.

(5) Function Loading

Use the proctor command:

```
SC   FNLDR
```

to load the relocatable binary deck onto the system. The function should now be ready for author and student use.

2. FUNCTION TESTING

Put the course CALCT on one of the course packs with the card assembler. This course is designed to call the function using all of the possible parameters. It should be readily evident, through following this course, whether all parts of the function are working properly.

III. FUNCTION FCALC PROGRAM LOGIC

III. FUNCTION FCALC PROGRAM LOGIC

Two physical and three logical sections make up Coursewriter II function FCALC. The first physical section is relocatable. In use, it is loaded by the system into one of the function areas. The second physical section is an absolute assembly which is loaded into the application two area.

The three logical sections are the loader, the interpreter/evaluator, and the output converter. Sections 1 and 3, the loader and the output converter are in the relocatable portion. The interpreter/evaluator comprises the absolute portion. The program flow of each of the logical sections will be discussed separately in the following paragraphs.

1. THE ABSOLUTE TEXT LOADER

The first portion of the relocatable section is a loader. Its purpose is to insure that the interpreter/evaluator is in core and to transfer to it. It accomplishes this by checking the two words directly preceding the application two area. If they are the correct word count and sector address respectively, of the fixed portion, it is assumed that the fixed portion is in core. If not, the two words are loaded with the correct word count and sector address. Transfer is then made to GGET to load the

absolute text from the system disk. As soon as it is established that the proper text is in core, transfer is made to it.

2. THE INTERPRETER/EVALUATOR

The interpreter/evaluator scans buffer 0, a character at a time. Each character is evaluated to determine if it is an operator, an operand portion, the beginning of a function call, a parenthesis, a character to be ignored, or an illegal character. Appropriate action is then taken. Operators and operands are placed in separate stacks. Function calls generate operators and sometimes dummy operands. Error calls are made if any syntax errors are discovered. The scan ends when an equal sign or an EOB is encountered. The TRVRS routine then traverses the operator stack, and using the appropriate operands, returns a numeric result in floating point format. If the input parameter indicates that a result is to be returned, control is then transferred to the output converter. If not, buffer 5 is scanned in the same manner as buffer 0. Comparison is then made of the two results with the CMPRE routine. Counters 22 and 23 are set to appropriate values, and control is returned to the relocatable text.

3. THE OUTPUT CONVERTER

The output converter converts a floating point number contained in RSLT1 to display code scientific notation. That is, all numbers will be converted to $n \times 10^p$ where $1.000 \leq |n| \leq 9.999$ and $-38 \leq p \leq +38$, $p \neq 0$.

If p - 0, only n is put into the buffer. These characters will be in a P = 0 form suitable for display with the DT operation code.

4. OPERATIONAL SUBROUTINES

Many subroutines are used to accomplish the purposes of the function. Each of them will be discussed separately. They are listed in the order in which they appear in the listings.

(1) Subroutine PROG

This subroutine contains the entry point to the absolute text. It evaluates the input parameter and sets appropriate flags. It obtains the locations of the input buffers and transfers them to subroutine SETUP. After evaluating buffer 0 it stores the result in RSLT1. Control is then returned to the relocatable text if the input parameter is 3 or 4. If not, it evaluates buffer 5, and calls subroutine CMPRE to compare the results of the two computations. Control is then returned to the relocatable text.

- Calling instruction -- BSI L /6802
- Transfer instruction -- BSC I /6802
- Subroutines called -- BUILD, CMPRE, SETUP, TRVRS

(2) Subroutine BUILD

This subroutine is used to build the operator and operand stacks. It scans an input buffer, checks syntax, and transfers to appropriate subroutines depending on the characters encountered. It returns upon encountering an equal sign or an EOB.

- Calling instruction -- BSI L BUILD
- Transfer instruction -- BSC L BLD11
- Subroutines called -- BLD01, BLD02, BLD03, BLD04, BLD05, BLD06, BLD07, BLD08, BLD09, BLD10, BLD11, BLD12, BLD13, BLD14, GTNXT

(3) Subroutine BLD12

The use of the alphabetic character "x" is handled by this subroutine. If the calling parameter is 2, "x" is handled as an algebraic symbol. Otherwise, it is handled as a times sign.

- Calling instruction -- BSC L BLD12
- Transfer instruction -- in line code
- Subroutines called -- BLD13

(4) Subroutine BLD13

Algebraic symbols are handled by this subroutine. If the calling parameter is not 2, transfer is made to subroutine ERROR

with parameter of 9. If the calling parameter is 2, the symbol is converted into a numeric value:

- Calling instruction -- BSC L BLD13
- Transfer instruction -- BSC L BLD10
- Subroutines called -- BLD10, ERROR

(5) Subroutine BLD01

This subroutine handles the minus sign. The minus sign may be the sign of an operand or an operator. If it is the sign of an operand, a flag is set. If it is placed as a monadic operator, it is converted to diadic by placing a zero operand on the operand stack. Operator 1 is then placed on the operator stack.

- Calling instruction -- BSC L BLD01
- Transfer instruction -- BSC L A1
- Subroutines called -- BAKUP, ERROR, GTNXT, ORAND, ORTOR

(6) Subroutine BLD02

This subroutine handles the plus operator. If it is used as a monadic operator, it converts it to diadic by adding a zero operand to the operand stack. Operator 2 is then placed on the operator stack.

- Calling instruction -- BSC L BLD02
- Transfer instruction -- BSC L A1
- Subroutines called -- ERROR, ORAND, ORTOR

(7) Subroutine BLD03

Multiply operators are handled by this subroutine. Operator 3 is placed in the operator stack.

- Calling instruction -- BSC L BLD03
- Transfer instruction -- BSC L A1
- Subroutines called -- ERROR, ORAND, ORTOR

(8) Subroutine BLD04

Divide operators are handled by this subroutine. Operator 4 is placed in the operator stack.

- Calling instruction -- BSC L BLD04
- Transfer instruction -- BSC L A1
- Subroutines called -- ERROR, ORAND, ORTOR

(9) Subroutine GTNXT

This subroutine is used to obtain the next character in the buffer. By use of other subroutines, it obtains the next valid character but ignores backspaces, blanks, pairs of index, reverse index, and pairs of reverse index, index.

- Calling instruction -- BSI L GTNXT
- Transfer instruction -- BSC I GTNXT
- Subroutines called -- GTCHR, PAIRL, PAIRR

(10) Subroutine GTCHR

The next character in the input buffer is obtained by subroutine GTCHR. It is returned in the AC, right adjusted, zero filled. Blanks and backspaces are ignored by subroutine GTCHR.

- Calling instruction -- BSI L GTCHR
- Transfer instruction -- BSC I GTNXT
- Subroutines called -- none

(11) Subroutine PAIRL

This subroutine is used to see if the character in the AC is an index. If it is not, the subroutine returns. If it is, check is made for the next character. If this character is a reverse index, both characters are ignored, and a new character is obtained by use of GTCHR. The test is continued until all sequential pairs of index, reverse index are bypassed.

- Calling instruction -- BSI L PAIRL
- Transfer instruction -- BSI I PAIRL
- Subroutines called -- GTCHR

(12) Subroutine PAIRR

The same function as PAIRL is performed by PAIRR, except that PAIRR checks for reverse index, index pairs.

- Calling instruction -- BSI L PAIRL
- Transfer instruction -- BSC I PAIRL
- Subroutines called -- PAIRR

(13) Subroutine BAKUP

The pointers which are used by GTCHR are decremented up by BAKUP to point to the next character which is not a blank or a backspace.

- Calling instruction -- BSI L BAKUP
- Transfer instruction -- BSC I BAKUP
- Subroutines called -- none

(14) Subroutine CONV

The four integer values held in CRHLD through CRHLD+3 are used as decimal digits to create a single integer value. The value in POINT is used as a power of ten by which to multiply this integer value. Subroutine CONV uses these values to produce a floating point number returned in the AC and Q.

- Calling instruction -- BSI L CONV
- Transfer instruction -- BSC I CONV
- Subroutines called -- FLDR, FLOAT, FMPY, PACKR, SCHR, SWAP

(15) Subroutine GTOPR

The next operator in the operator stack is obtained by GTOPR. If there are none left in the stack, GTOPR returns with -1 in the AC.

- Calling instruction -- BSI L GTOPR
- Transfer instruction -- BSC I GTOPR
- Subroutines called -- none

(16) Subroutine ORTOR

The value in the AC, ored with LEVEL is placed on the operator stack by ORTOR.

- Calling instruction -- BSI L ORTOR
- Transfer instruction -- BSC I ORTOR
- Subroutines called -- none

(17) Subroutine ORAND

The value in the AC and Q is placed on the operand stack by ORAND.

- Calling instruction -- BSI L ORAND
- Transfer instruction -- BSC I ORAND
- Subroutines called -- ERROR

(18) Subroutine PTNST

This subroutine places the current operation level, found in LEVEL, on the nest stack.

- Calling instruction -- BSI L PTNST
- Transfer instruction -- BSC I PTNST
- Subroutines called -- ERROR

(19) Subroutine GTNST

This subroutine obtains the top entry from the nest stack.

- Calling instruction -- BSI L GTNST
- Transfer instruction -- BSC I GTNST
- Subroutines called -- none

(20) Subroutine BKNST

The pointer used by GTNST is backed up one position by subroutine BKNST.

- Calling instruction -- BSI L BKNST
- Transfer instruction -- BSC I BKNST
- Subroutines called -- none

(21) Subroutine CMPRE

The floating point values contained in RSLT1 and the AC and Q are compared according to the formula shown in paragraph I.1.(1). If the values match, 0 is stored indirectly in ATR22 and ATR23. If not, -1 is stored indirectly in ATR22 and ATR23.

- Calling instruction -- BSI L CMPRE
- Transfer instruction -- BSC I CMPRE
- Subroutines called -- FABS, FLDL, FLDR, FMPY, FSUB, PACKR

(22) Subroutine TRVRS

The operator and operand stacks are used by TRVRS to calculate a value for an expression. This floating point value is returned in the AC and Q.

- Calling instruction -- BSI L TRVRS
- Transfer instruction -- BSC I TRVRS
- Subroutines called -- CALC, GTOPR

(23) Subroutine CALC

The operator indicated by SVEOP is obtained from the operator stack by CALC. The appropriate operands are then loaded into LEFT and RIGHT. By using the floating point routine indicated by the operator, a floating point result is returned in the AC and Q.

- Calling instruction -- BSI L CALC
- Transfer instruction -- BSC I CALC
- Subroutines called -- FADD, FAXB, FDIV, FLDL, FLDR, FLOG, FMPY, FSUB, PACKR

(24) Subroutine BLD14

An alphabetic 0 is converted by BLD14 into a numeric 0 for use by BLD10.

- Calling instruction -- BSC L BLD14
- Transfer instruction -- in line code
- Subroutines called -- none

(25) Subroutine BLD10

This subroutine is used to convert one or more decimal digits in the input stream into a floating point number which is placed on the operand stack.

- Calling instruction -- BSC L BLD10
- Transfer instruction -- BSC L A1
- Subroutines called -- BAKUP, CONV, ERROR GTNUM, ORAND, ORTOR

(26) Subroutine GTNUM

The character in the AC is evaluated by GTNUM. If it is an EBCDIC decimal point, it is returned intact. If it is an EBCDIC numeric digit, it is converted into a binary integer. If it is anything else, a -1 is returned.

- Calling instruction -- BSI L GTNUM
- Transfer instruction -- BSC I GTNUM
- Subroutines called -- BAKUP, GTNXT

(27) Subroutine BLD05

This subroutine is called when an "1" is encountered in the input buffer. The routine then scans for the characters "og" immediately following the "1". If they are found, a dummy operand is placed on the operand stack, and an operator 6 is placed on the operator stack. Otherwise, the "1" is replaced by an EBCDIC 1 and control is transferred to BLD10.

- Calling instruction -- BSC L BLD05
- Transfer instruction -- BSC L A1
- Subroutines called -- BAKUP, ERROR, GTNXT, ORAND, ORTOR, PTNST

(28) Subroutine BLD06

Each time a left parenthesis is encountered, BLD06 increments the current level of operation, found in LEVEL. It loads an operator 3 in the operator stack if the preceding character was part of an operand.

- Calling instruction -- BSC L BLD06
- Transfer instruction -- BSC L A1
- Subroutines called -- ORTOR

(29) Subroutine BLD07

Each time a right parenthesis is encountered, BLD07 decrements LEVEL. If the decremented level matches the last entry in the level nest, LEVEL is decremented again.

- Calling instruction -- BSC L BLD07
- Transfer instruction -- BSC L A1
- Subroutines called -- BKNST, ERROR, GTNST

(30) Subroutine BLD08

Operator number 5 is loaded into the operator stack by BLD08, and LEVEL is incremented.

- Calling instruction -- BSC L BLD08
- Transfer instruction -- BSC L A1
- Subroutines called -- ERROR, ORTOR

(31) Subroutine BLD09

The current level of operation, found in LEVEL, is decremented by BLD09.

- Calling instruction -- BSC L BLD09
- Transfer instruction -- BSC L A1
- Subroutines called -- ERROR

(32) Subroutine BLD11

Whenever an EOB or an equal sign is encountered, BLD11 checks the current level. If LEVEL is incorrect, transfer is made to ERROR. If not, a return transfer is made from the BUILD subroutine.

- Calling instruction -- BSC L BLD11
- Transfer instruction -- BSC I BUILD
- Subroutines called -- ERROR

(33) Subroutine SETUP

This subroutine initializes all flags and switches which are necessary for evaluating the expression in a buffer.

- Calling instruction -- BSI L SETUP
- Transfer instruction -- BSC I SETUP
- Subroutines called -- none

(34) Subroutine ERROR

Transfer is made to this subroutine on any syntax or arithmetic error. The error number is stored indirectly in ATR23 and the current character number $\pm$ 1 is stored indirectly in ATR22.

- Calling sequence -- BSI L ERROR
  DC ERNUM
- Transfer instruction -- BSC I PROG
- Subroutines called -- none

5. FLOATING POINT SUBROUTINES

Since the 1130 and 1800 computers have no floating point hardware, all floating point operations are done through subroutines. Two floating point formats are used by these routines. The first format uses three words to hold a floating point number. In this format, the exponent is contained in the low order 8 bits of the first word. The next word, on an

even word boundary, contains the high order 16 bits of the fraction, and the last word contains the low order 16 bits of the fraction. The labels RIGHT and LEFT address such three word aligned locations to hold floating point operands. The second floating point format is two words long. It is used generally for storage of floating point values. In this format the high order 24 bits of the fraction are left adjusted in two succeeding words. The exponent is contained in the remaining 8 bits.

Each of the following floating point subroutine descriptions indicates the format used. It is assumed that the index registers are set before a call is made to any floating point routine. Index register 1 should contain the address of RIPNT, index register 2 should contain the address of R2PNT, and index register 3 should contain the address of R3PNT.

(1) Subroutine FAXB

This subroutine takes the number in LEFT and raises it to the power in RIGHT. The result is stored in RIGHT. The evaluation formula and error values are found in Exhibit I-1.

- Calling instruction -- BSI 1 FAXB-RIPNT
- Transfer instruction -- BSC I FAXB
- Subroutines called -- ERROR, FABS, FDIV, FLDL, FLDR, FLN, FMPY, FXPN, IFIX, PACKR, SCHR, SWAP

(2) Subroutine PACKR

The 32 bit fraction in RIGHT is rounded to 24 bits. These 24 bits are left justified in the AC and Q. The remaining 8 bits are replaced by the exponent which is found in RIGHT -1.

- Calling instruction -- BSI 1 PACKR-R1PNT
- Transfer instruction -- BSC I PACKR
- Subroutines called -- FRNG, SAVER, SCHR

(3) Subroutine FABS

A call to this subroutine results in the 3 word floating point number in RIGHT-1 through RIGHT+1 being converted to its absolute value.

- Calling instruction -- BSI 1 FABS-R1PNT
- Transfer instruction -- BSC I FABS
- Subroutines called -- SCHR

(4) Subroutine FLOAT

This subroutine converts the two word integer in the AC and Q into a 3 word floating point number which is stored in RIGHT.

- Calling instruction -- BSI 1 FLOAT-R1PNT
- Transfer instruction -- BSC I FLOAT
- Subroutines called -- NORM

(5) Subroutine FLOG

The logarithm base 10 of the value in RIGHT is returned to RIGHT by subroutine FLOG. The value in LEFT is destroyed.

- Calling instruction -- BSI 1 FLOG-R1PNT
- Transfer instruction -- BSC I FLOG
- Subroutines called -- FDIV, FLDR, FLN, SWAP

(6) Subroutine FSUB

The value in RIGHT is subtracted from the value in LEFT by subroutine FSUB. The result replaces the value in RIGHT.

- Calling instruction -- BSI 2 FSUB-R2PNT
- Transfer instruction -- in line to FADD
- Subroutines called -- FRNG, LOADL, MOVEL, NORM, SAVEL, SCHR, SWAP

(7) Subroutine FADD

The value in RIGHT is added to the value in LEFT by subroutine FADD. The result replaces the value in RIGHT.

- Calling instruction -- BSI 2 FADD-R2PNT
- Transfer instruction -- BSC I FSUB
- Subroutines called -- FRNG, LOADL, MOVEL, NORM, SAVEL, SCHR, SWAP

(8) Subroutine SAVER

This subroutine places the value in RIGHT into RSAVE.

- Calling instruction -- BSI 2 SAVER-R2PNT
- Transfer instruction -- BSC I SAVER
- Subroutines called -- none

(9) Subroutine SAVEL

This subroutine places the value in LEFT into LSAVE.

- Calling instruction -- BSI 2 SAVEL-R2PNT
- Transfer instruction -- BSC I SAVEL
- Subroutines called -- SAVER

(10) Subroutine FMPY

The value in LEFT is multiplied by the value in RIGHT. The result replaces the value in RIGHT.

- Calling instruction -- BSI 2 FMPY-R2PNT
- Transfer instruction -- BSC I FMPY
- Subroutines called -- FRNG, MPFRC, NORM

(11) Subroutine MPFRC

This subroutine multiplies the 32 bit fraction in RIGHT by the 32 bit fraction in the AC and Q. The rounded result is returned in the AC and Q.

- Calling instruction -- BSI 2 MPFRC-R2PNT
  or
  BSI 2 XMDS-R2PNT
- Transfer instruction -- BSC I MPFRC
- Subroutines called -- none

(12) Subroutine SWAP

The values in LEFT and RIGHT are switched by this subroutine.

- Calling instruction -- BSI 2 SWAP-R2PNT
- Transfer instructions -- BSC I SWAP
- Subroutines called -- none

(13) Subroutine MOVER

This subroutine shifts the fraction in RIGHT to align the implied decimal point according to the biased exponent contained in the AC.

- Calling instruction -- BSI 2 MOVER-R2PNT
- Transfer instruction -- BSC I MOVER
- Subroutines called -- none

(14) Subroutine MOVEL

This subroutine shifts the fraction in LEFT to align the implied decimal point according to the biased exponent contained in the AC.

- Calling instruction -- BSI 2 MOVEL-R2PNT
- Transfer instruction -- BSC I MOVER
- Subroutines called -- none

(15) Subroutine FRNG

This subroutine checks the biased exponent contained in RIGHT-1. If it is negative, floating underflow is indicated. If it is greater than 255, floating point overflow is indicated. On either condition, a transfer to the ERROR routine is made. If no error is found, the subroutine returns.

- Calling instruction -- BSI 2 FRNG-R2PNT
- Transfer instruction -- BSC I FRNG
- Subroutines called -- ERROR

(16) Subroutine NORM

The three word floating point number in RIGHT is normalized by subroutine NORM. That is, the fraction is shifted, if necessary,

so that it is either 0 or greater than 1/2. The exponent is adjusted to reflect the shift.

- Calling instruction -- BSI 2 NORM-R2PNT
- Transfer instruction -- BSC I NORM
- Subroutines called -- MOVER

(17) Subroutine SCHR

The sign of the number held in RIGHT is changed by subroutine SCHR.

- Calling instruction -- BSI 2 SCHR-R2PNT
- Transfer instruction -- BSCI SCHR
- Subroutines called -- none

(18) Subroutine FDIV

This subroutine replaces the contents of RIGHT with the quotient of the value in LEFT divided by the value in RIGHT.

- Calling instruction -- BSI 2 FDIV-R2PNT
- Transfer instruction -- BSC I FDIV
- Subroutines called -- FRNG, NORM

(19) Subroutine IFIX

The value in RIGHT is rounded to the nearest integer. It is then converted to integer format and stored in RIGHT and in the AC.

- Calling instruction -- BSI 3 IFIX-R3PNT
- Transfer instruction -- BSC I IFIX
- Subroutines called -- FABS, FADD, FLDR, SWAP

(20) Subroutine LOADR

The value in SAVER is put in RIGHT by subroutine LOADR.

- Calling instruction -- BSI 3 LOADR-R3PNT
- Transfer instruction -- BSC I LOADR
- Subroutines called -- none

(21) Subroutine LOADL

The value in SAVEL is put in LEFT by subroutine LOADL.

- Calling instruction -- BSI 3 LOADL-R3PNT
- Transfer instruction -- BSC I LOADL
- Subroutines called -- LOADR

(22) Subroutine FXPN

A polynomial approximation of e to the power in RIGHT is made by FXPN. The result is stored in RIGHT.

- Calling instruction -- BSI 3 FXPN-R3PNT
- Transfer instruction -- BSC I FXPN
- Subroutines called -- ERROR, FLDR, FRNG, MPFRC

(23) Subroutine FLN

A polynomial approximation of the natural logarithm of the value in RIGHT is made by FLN. The result is stored in RIGHT.

- Calling instruction -- BSI 3 FLN-R3PNT
- Transfer instruction -- BSC I FLN
- Subroutines called -- FADD, FDIV, FLDR, FLOAT, FMPY, FSUB, LOADL, MOVEL, SAVEL, SAVER, SWAP, MPFRC

(24) Subroutine FLDR

The two word floating point number in the AC and Q is converted into three word format and stored in RIGHT-1 through RIGHT+1 by subroutine FLDR.

- Calling instruction -- BSI 3 FLDR-R3PNT
- Transfer instruction -- BSC I FLDR
- Subroutines called -- none

(25) Subroutine FLDL

The two word floating point number in the AC and Q is converted into three word format and stored in LEFT-1 through LEFT+1 by subroutine FLDL.

- Calling instruction -- BSI 3 FLDL-R3PNT
- Transfer instruction -- BSC I FLDL
- Subroutines called -- FLDR

IV. PROGRAM MODIFICATION NOTES

IV. PROGRAM MODIFICATION NOTES

Modification of the Coursewriter II function FCALC should be attempted only by programmers who are thoroughly familiar with the Coursewriter II operating system as well as with the program logic of the function.

1. LABEL RELOCATION

This procedure should be followed whenever any part of the absolute text is modified. The relocatable portion of the function makes many references, through EQUs, to labels in the absolute portion of the function. Hence, any additions or deletions made to the coding of the absolute portion will require that the EQU table in the relocatable portion be suitably modified.

2. RELATIVE ERROR

The label FUDGE in the absolute portion addresses the location containing the allowable relative error in buffer 5/buffer 0 comparison. The statement is now:

```
FUDGE   DEC   .01   RELATIVE ERROR.
```

Change of this statement to any desired value may be made. Reassembly of the absolute portion of the function will be necessary as well as

following of part 4 of the installation guide to effect the change.

Alternatively, the binary deck may be patched by making up a card as follows:

C6A44 XXXX XXXX *CHANGE RELATIVE ERROR ALLOWANCE.

XXXX XXXX is to be replaced by the relative error in hexadecimal 1800/1130 floating point format. For example, if XXXX XXXX is replaced by 68DB 8B73, the relative error allowance figure will be 0.0001. Patching the binary deck will eliminate the need for reassembly of the absolute text.

3. NEW ALGORITHM ADDITION

The following procedure should be used to include other mathematical algorithms such as SIN, COS, TAN, et cetera in function FCALC. Note that there are three parts which comprise an algorithm (function) call. They are: first, the first character recognition routine, second, the name recognition routine, and third, the calculation routine.

(1) First Character Recognition

Immediately preceding the label MRUTE, place:

DC /00XX

where XX is replaced by the EBCDIC representation of the first character in the name. Names should not begin with x, 1, or o.

Considerable difficulty will be encountered in using these characters as first characters in names, since these characters are considered as times sign, one, and zero respectively, and special check is made for them by other portions of the program.

(2) First Character Transfer

Immediately following the last constant of the test and transfer tables, place:

```
DC    MMMMM
```

where MMMMM is the label of the first word of the name recognition routine.

(3) Minus Operator Modification

Modify routine BLD01 so that it recognizes the first character of the algorithm name and treats it the same as it does a "(" or an "1".

(4) Name Recognition Routine

Build a routine to use GTNXT and other available subroutines to recognize the name of the function call. It may be patterned after routine BLD05 which recognizes the name "log". If the function is monadic, as log is, be sure to use subroutine ORAND to put a

dummy operand in the operand table. Put an operator in the operator table with subroutine ORTOR. Assign new operator numbers starting with seven.

This routine and the calculation routine should be placed at the end of the absolute text to prevent short instruction linkage interference with other routines.

(5) Calculation Routine

Immediately preceding label BLD10, put:

```
DC   XXXXX
```

where XXXXX is to be replaced by the label beginning the calculation routine. Construct a routine to use the floating point subroutines to provide the necessary calculation. Assume arguments are contained in LEFT and RIGHT. If function is monadic, argument will be in RIGHT. The result should be returned to RIGHT.

(6) Implementation

Reassemble the absolute program. If the length of the absolute text exceeds 6 sectors (1920 words), the ABC card must be modified to reflect the new length. Additionally, the statement:

```
NMSEK   EQU   6
```

in the relocatable portion must be modified to match the ABC card. Follow the Label Relocation procedure and reassemble the relocatable text. Follow the installation guide to effect the change.

APPENDIX A

Source Listing for Coursewriter II Function FCALC

```
// JOB
// ASM
*LIST
*PRINT SYMBOL TABLE
*                    PROG
*
        ABS
APPL2   EQU     /6802
STPLC   EQU     APPL2
        ORG     STPLC
PROG    DC      0
        LDX   L1 R1PNT          SET UP XR1
        LDX   L2 R2PNT
        LDX   L3 R3PNT          SET UP XR3
*
*                    PARAMETER
*
*                    %0 IS NON-INTEGER EXPONENTS AND NO
*                       RESULT RETURNED
*                    %1 IS INTEGER EXPONENTS AND NO RESULT
*                       RETURNED
*                    %2 IS ALGEBRAIC INPUT AND INTEGER
*                       EXPONENTS WITH NO RESULT RETURNED
*                    %3 IS NON-INTEGER EXPONENTS WITH
*                       RESULT RETURNED TO BUFFER 0
*                    %4 IS INTEGER EXPONENTS WITH RESULT
*                       RETURNED TO BUFFER 0
*
        SLA     16
        STO   2 IFLG-R2PNT      RESET INTEGER FLAG
        STO   3 AFLG-R3PNT      RESET ALGEBRAIC FLAG
        STO   3 RFLG-R3PNT      RESET RETURN FLAG
        LD    I ATRE6           FUNCTION PARAMETER
        SLA     4               DELETE /F
        SRA     12
        S     2 K1-R2PNT
        BSC     Z+
        MDX     B0000
        S     2 K1-R2PNT
        BSC     Z+
        MDX     B0001
        S     2 K1-R2PNT
        BSC     Z+
        MDX     B0002
        S     2 K1-R2PNT
        BSC     Z+
        MDX     B0003
        LD    2 HFFFF-R2PNT
        STO   2 IFLG-R2PNT
B0004   STO   3 RFLG-R3PNT
        MDX     B0000
B0001   STO   2 IFLG-R2PNT
        MDX     B0000
B0002   STO   3 AFLG-R3PNT
        MDX     B0001
B0003   MDX     B0004
B0000   EQU     *
        LD      BOAD            LOAD ADDR OF BUFFER 0
        BSI   L SETUP           SETUP CONSTANTS
        BSI     BUILD           BUILD OPERATOR AND
*                               OPERAND TABLES
```

```
        BSI  L  TRVRS           USE TABLES TO OBTAIN    CF000332
*                               RESULT                  CF000333
        STD   3 RSLT1-R3PNT                             CF000342
        MDX  L  RFLG,0                                  CF000345
        MDX     PRTRN           RESULT RETURN FLAG ON   CF000348
        LD      BSAD            POINT TO BUFFER 5       CF000351
        BSI  L  SETUP                                   CF000360
        BSI     BUILD                                   CF000370
        BSI  L  TRVRS                                   CF000380
        BSI  L  CMPRE                                   CF000390
PRTRN   BSC  I  PROG                                    CF000400
ATR     EQU     /1D7C           TERMINAL RECORD AREA    CF000410
ATRB0   EQU     ATR+35          BUFFER 0 ADDRESS        CF000420
ATRB5   EQU     ATR+285         BUFFFR 5 ADDRESS        CF000430
ATRE6   EQU     ATR+370         ADDRESS OF FN PARAMETRSCF000440
*                               AMETER                  CF000450
ATRC0   EQU     ATR+489         COUNTER 0               CF000460
ATR22   EQU     ATRC0+22        COUNTER 22              CF000470
ATR23   EQU     ATRC0+23        COUNTER 23              CF000480
B0AD    DC      ATRB0                                   CF000490
B5AD    DC      ATRB5                                   CF000500
*                                                       CF000510
*                                                       CF000520
*               BUILD                                   CF000530
*                                                       CF000540
BUILD   DC      0                                       CF000550
A1      LD      CONTB           SET UP CONSTANT AND     CF000560
        STO     CONPK           ROUTINE POINTERS        CF000570
        STO     CNPK2                                   CF000580
        LD      RUTAB                                   CF000590
        STO     TRPNT                                   CF000600
        BSI  L  GTNXT            GET NEXT CHARACTER     CF000610
        S     3 C0-R3PNT                                CF000620
        BSC     +Z                                      CF000630
        MDX     NONUM                                   CF000640
        A     3 C0-R3PNT                                CF000650
        BSC  L  BLD10                                   CF000660
*               BRANCH IF A NUMBER                      CF000670
NONUM   A     3 C0-R3PNT                                CF000680
CONPK   EQU     *+1                                     CF000690
NXTCN   EOR  L  *-*             EOR WITH CONSTANT       CF000700
TRPNT   EQU     *+1                                     CF000710
        BSC  I  *-*,+-          BRANCH TO ITS ROUTINE ICF000720
*                               IF ZERO                 CF000730
CNPK2   EQU     *+1                                     CF000740
        EOR  L  *-*             EOR WITH SAME CONSTANT  CF000750
        SRT     16              PUT IN Q                CF000760
        MDX  L  CONPK,1         MODIFY POINTERS         CF000770
        MDX  L  CNPK2,1                                 CF000780
        MDX  L  TRPNT,1                                 CF000790
        LD      CONPK                                   CF000800
        S       RUTAB                                   CF000810
        BSC     -               ERROR IF ZERO           CF000820
        MDX     BLD13                                   CF000830
        SLT     16              PUT BACK IN AC          CF000840
        MDX     NXTC                                    CF000850
*                                                       CF000860
*                                                       CF000870
BLD12   MDX  L  AFLG,0                                  CF000880
        MDX     B1200           BRANCH IF ON            CF000890
        BSC  L  BLD03           CONSIDER AS TIMES SIGN  CF000900
```

```
B1200 EOR  I  CONPK                                                  CF000830
      SRT     16                  PUT IN 0 FOR SHIFT BACK            CF000840
*                                                                    CF000850
*             BLD13                                                  CF000860
*             HANDLES ALGEBRAIC SYMBOLS                              CF000870
*                                                                    CF000880
BLD13 MDX  L  AFLG,0              TEST ALGEBRAIC FLAG                CF000890
      MDX     B1301               FLAG IS ON                         CF000900
      BSI  L  ERROR               FLAG OFF--SYMBOL IS                CF000910
      DC      9                   ILLEGAL                            CF000920
H00F0 DC      /00F0                                                  CF000930
B1301 SLT     16                  PUT BACK IN AC                     CF000940
      OR      H00F0               TURN INTO PSEUDO-NUMERI            CF000950
      A     2 K2-R2PNT            ADD 2 TO INTEGER                   CF000960
      STO   3 NFLG-R3PNT                                             CF000970
      BSC  L  BLD10               GO TO NUMERIC ROUTINE              CF000980
*                                                                    CF000990
*             TEST AND TRANSFER TABLES                               CF001000
*                                                                    CF001010
RUTAB DC      MRUTE                                                  CF001020
CONTR DC      MINUS                                                  CF001030
MINUS DC      /0060                                                  CF001040
PLUS  DC      /004E                                                  CF001050
CAPX  DC      /00E7                                                  CF001060
SMALX DC      /00A7                                                  CF001070
TIMES DC      /00B1                                                  CF001080
SLASH DC      /0061                                                  CF001090
DIVID DC      /00B2                                                  CF001100
SMALL DC      /0093                                                  CF001110
CAPL  DC      /00D3                                                  CF001120
LPREN DC      /004D                                                  CF001130
CAPO  DC      /00D6                                                  CF001140
SMALO DC      /0096                                                  CF001150
DCPNT DC      /004B                                                  CF001160
RPREN DC      /005D                                                  CF001170
UP    DC      /0009                                                  CF001180
DOWN  DC      /0025                                                  CF001190
EOBB  DC      /0026                                                  CF001200
EQUAL DC      /007E                                                  CF001210
CR    DC      /0015                                                  CF001220
CONT  DC      /0027                                                  CF001230
MRUTE DC      BLD01               MINUS                              CF001240
      DC      BLD02               PLUS                               CF001250
      DC      BLD12               X OR TIMES                         CF001260
      DC      BLD12               X OR TIMES                         CF001270
      DC      BLD03               TIMES                              CF001280
      DC      BLD04               DIVIDE                             CF001290
      DC      BLD04               DIVIDE                             CF001300
      DC      BLD05               LOG                                CF001310
      DC      BLD05               LOG                                CF001320
      DC      BLD06               LEFT PARENTHESIS                   CF001330
      DC      BLD14               TREAT ALPHA O AS 0                 CF001340
      DC      BLD14               TREAT ALPHA O AS 0                 CF001350
      DC      BLD10               NUMERIC                            CF001360
      DC      BLD07               RIGHT PARENTHESIS                  CF001370
      DC      BLD08               POWER BEGIN                        CF001380
      DC      BLD09               POWER END                          CF001390
      DC      BLD11               EOB                                CF001400
      DC      BLD11               EOB                                CF001410
      DC      A1                  CARRIAGE RUTURN                    CF001420
      DC      A1                  CONTINUE                           CF001430
```

```
*                                                              CF001260
*                                                              CF001290
*            BLD01                                             CF001300
*                                                              CF001310
*            TAKES CARE OF MINUS OPERATOR                      CF001320
*                                                              CF001330
BLD01 LD     3 PREC-R3PNT                                      CF001340
      BSC      +-                                              CF001350
      MDX      B0100           ERROR OP OP SEQ                 CF001360
      BSC      -                                               CF001370
      MDX      B0101                                           CF001375
      MDX   L  MFLAG,0                                         CF001377
      MDX      B0100                                           CF001378
      BSI   L  GTNXT           AT EXPRESSN BEGINNING           CF001381
      STO   2  STAT-R2PNT      SAVE CHARACTER                  CF001382
      BSI   L  BAKUP           RESET GTNXT                     CF001383
      LD    2  STAT-R2PNT      RESTORE CHARACTER               CF001384
      EOR   3  LPRN-R3PNT      CHECK FOR LEFT PAREN            CF001385
      BSC   L  B0103,+-        YES -- PUT IN 0 OPERAND         CF001386
      EOR   3  LPRN-R3PNT      NO                              CF001387
      AND   3  LMASK-R3PNT     CHANGE TO LOWER CASE            CF001388
      EOR   3  LITL-R3PNT      CHECK FOR LOG                   CF001389
      BSC   L  B0103,+-        YES -- PUT IN 0 OPERND          CF001390
      STO   3  MFLAG-R3PNT     NO SET SIGN FLAG                CF001391
      MDX      A1              RETURN                          CF001392
B0103 SLT      32              ZERO OPERAND                    CF001393
      BSI   L  ORAND           DUMMY OPERAND ON STACK          CF001400
B0101 LD     2  K1-R2PNT       %1 CALLS FOR MINUS              CF001410
B0102 BSI   L  ORTOR           PUT ON OPERATOR STACK           CF001420
      SLA      16              ZERO AC                         CF001430
      STO   3  PREC-R3PNT                                      CF001440
      BSC   L  A1              RETURN FOR NEXT CHARAC          CF001450
B0100 BSI   L  ERROR                                           CF001460
      DC       7                                               CF001470
*                                                              CF001480
*            BLD02 TAKES CARE OF PLUS OPERATOR                 CF001490
*                                                              CF001500
*                                                              CF001510
BLD02 LD     3 PREC-R3PNT                                      CF001520
      BSC      +-                                              CF001530
      MDX      B0100           ERROR OP-OP SEQUENCE            CF001540
      BSC      -                                               CF001550
      MDX      B0201                                           CF001560
      SLT      32              ZERO AC + Q PUT DUMMY           CF001570
      BSI   L  ORAND           OPERAND ON STACK                CF001580
B0201 LD     2  K2-R2PNT       PUT + OP                        CF001590
      MDX      B0102           ON OPERATOR STACK               CF001600
*                                                              CF001610
*            BLD03                                           CF001620
*            MULT OPERATOR                                   CF001630
*                                                              CF001640
BLD03 LD     3 PREC-R3PNT                                      CF001650
      BSC      +-                                              CF001660
      MDX      B0100           OP-OP SEQ                       CF001670
      BSC      +                                               CF001680
      MDX      B0300           ERROR 8  OPERATOR 1ST           CF001690
      LD    2  K3-R2PNT                                        CF001700
      MDX      B0102           PUT ON STACK                    CF001710
*                                                              CF001720
B0300 BSI   L  ERROR           DIADIC OPERATOR                 CF001730
*            FIRST IN SEQUENCE                                 CF001740
```

```
        DC      8                                                       CF001750
*                                                                       CF001760
*               BLD04                                                   CF001770
*               DIVIDE OP                                               CF001780
*                                                                       CF001790
BLD04   LD    3 PREC-R3PNT                                              CF001800
        BSC     +-                                                      CF001810
        MDX     B0100                                                   CF001820
        BSC     +                                                       CF001830
        MDX     B0300                                                   CF001840
        LD    2 K4-R2PNT        DIVIDE OP                               CF001850
        MDX     B0102           PUT ON STACK                            CF001860
*                                                                       CF001870
*                                                                       CF001880
*                                                                       CF001890
*               GTNXT                                                   CF001900
*               OBTAINS NEXT AVAILABLE CHARACTER                        CF001910
*                                                                       CF001920
*                                                                       CF001930
GTNXT   DC      0                                                       CF001940
        BSI     GTCHR           BLANKS AND BKSPCS IGNR                  CF001950
        BSI     PAIRL           LF FOLLOWED BY RLF                      CF001960
*                               IGNORE                                  CF001970
        BSI     PAIRR           RLF FOLLOWED BY LF                      CF001980
*                               IGNORE                                  CF001990
        BSC   I GTNXT                                                   CF002000
SP50    DC      *-*             END OF BUFFER + 1                       CF002010
LEFTC   DC      *-*             LEFT OR RIGHT CHARACTER                 CF002020
BLANK   DC      /0040                                                   CF002030
EOB     DC      /0026                                                   CF002040
BKSP    DC      /0016                                                   CF002050
        DC      /0009           REVERSE LINE FEED                       CF002060
LF      DC      /0025           LINE FEED                               CF002070
RLF     DC      /0009           REVERSE LINE FEED                       CF002080
*                                                                       CF002090
*               GET NEXT CHARACTER BUT IGNORE                           CF002100
*               BLANKS AND BACKSPACES                                   CF002110
*                                                                       CF002120
GTCHR   DC      0                                                       CF002130
        LD      SP50            END OF BUFFER + 1                       CF002140
        S       CURAD           CURRENT ADDRESS                         CF002150
        BSC   L CEND,+  BRANCH IF CURAD GE SP50                         CF002160
CURAD   EQU     *+1                                                     CF002170
        LD    L *-*             GET NEXT WORD                           CF002180
        RTE     16              PUT IN Q                                CF002190
        LD      LEFTC                                                   CF002200
        BSC     Z               ZERO MEANS GET LEFT CH                  CF002210
        SLT     8                                                       CF002220
        SLT     8                                                       CF002230
        AND   2 H00FF-R2PNT                                             CF002240
        RTE     16              PUT IN Q                                CF002250
        LD      LEFTC                                                   CF002260
        EOR   2 K1-R2PNT        %1                                      CF002270
        STO     LEFTC                                                   CF002280
        BSC   L NORMP,Z                                                 CF002290
        MDX   L CURAD,1                                                 CF002300
NORMP   RTE     16              PUT BACK IN AC                          CF002310
        EOR     BLANK                                                   CF002320
        BSC   L GTCHR+1,+-  IGNORE BLANK                                CF002330
        EOR     BLANK                                                   CF002340
        EOR     BKSP                                                    CF002350
```

```
        BSC   L   GTCHR+1,+-                                        CF002360
        EOR       BKSP                                              CF002370
GRTN    BSC   I   GTCHR           RETURN                            CF002380
CEND    LD        EOB             LOAD EOB                          CF002390
        MDX       GRTN                                              CF002400
*                                                                   CF002410
*                                                                   CF002420
*               PAIRL AND PAIRR IGNORE PAIRS OF LF+RLF              CF002430
*               OR RLF + LF                                         CF002440
*                                                                   CF002450
PAIRL   DC        0                                                 CF002460
        STX   L1  GSV1+1          SAVE XR1                          CF002470
        LDX   I1  GPNTR           ADDRESS OF LF                     CF002480
DOAGN   EOR     1 0               CHECK FOR LF                      CF002490
        BSC   L   LFFND,+- BRANCH IF FOUND                          CF002500
        EOR     1 0                                                 CF002510
GSV1    LDX   L1  *-*                                              CF002520
        BSC   I   PAIRL                                             CF002530
LFFND   BSI       GTCHR           GET NEXT CHARACTER                CF002540
        EOR     1 -1              SEE IF NEXT IS RLF                CF002550
        BSC   L   SOVER,+- IF SO, IGNORE BOTH                       CF002560
        BSI       BAKUP NO SO BACKUP AND                            CF002570
        LD      1 0               PICK UP LF AGAIN                  CF002580
        MDX       GSV1            THEN RETURN                       CF002590
SOVER   BSI       GTCHR GET NEXT CHARACTER 2 HAVE BEEN IGNCF002600
*               IGNORED                                             CF002610
        MDX       DOAGN                                             CF002620
GPNTR   DC        LF                                                CF002630
PAIRR   DC        0                                                 CF002640
        MDX   L   GPNTR,1           IGNORE RLF                      CF002645
        BSI       PAIRL             THEN LF PAIR                    CF002650
        MDX   L   GPNTR,-1                                          CF002660
        BSC   I   PAIRR                                             CF002670
*                                                                   CF002680
*                                                                   CF002690
*               BAKUP                                               CF002700
*               BACKS UP GTNXT                                      CF002710
*                                                                   CF002720
*                                                                   CF002730
BAKUP   DC        0                                                 CF002740
        LD        LEFTC                                             CF002750
        EOR     2 K1-R2PNT                                          CF002760
        STO       LEFTC                                             CF002770
        BSC   L   BRTN,+-           JUMP FOR NO DECREMENT           CF002772
        MDX   L   CURAD,-1          DECREMENT CURAD                 CF002774
BRTN    LD    I   CURAD             LOAD CURRENT WORD               CF002776
        SRT       16                PUT IN 0                        CF002778
        LD        LEFTC             SEE IF LEFT OR RIGHT            CF002780
        BSC       Z                 SKIP TO GET LEFT                CF002782
        SLT       8                 GET RIGHT                       CF002784
        SLT       8                                                 CF002786
        AND     2 H00FF-R2PNT       ZAP LEFTMOST 8 BITS             CF002788
        EOR       BLANK             IF BLANK FOUND                  CF002790
        BSC   L   BAKUP+1,+-        BACKUP AGAIN                    CF002792
        EOR       BLANK                                             CF002794
        EOR       BKSP              IF BACKSPACE FOUND              CF002796
        BSC   L   BAKUP+1,+-        BACKUP AGAIN                    CF002798
        BSC   I   BAKUP             RETURN                          CF002800
*                                                                   CF002810
*                                                                   CF002820
*       CONV                                                        CF002830
```

```
*                                                              CF002940
*                                                              CF002950
CONV   DC      0                                               CF002960
       LD      3 NMDIG-R3PNT                                   CF002970
       BSC   L NGT0,Z-                                        CF002980
       SLT     32              NUMBER IS ZERO, SO RETU         CF002990
       STO     3 MFLAG-R3PNT   SIGN FLAG                       CF003000
       BSC   I CONV                                           CF003010
NGT0   STX     1 C1SV+1                                        CF003020
       STX     2 C2SV+1                                        CF003030
       LDX   L1 CRHLD                                         CF003040
       LDX   I2 NMSIG          NUMBER OF SIG DIGITS            CF003050
       SRA     16                                             CF003060
CLOOP  M       C10             MULT BY 10                      CF003070
       SLT     16              PUT 0 INTO AC                   CF003080
       A       1 0             ADD FIRST OR NEXT DIGIT         CF003090
       MDX     1 1             POINT TO NEXT                   CF003100
       MDX     2 -1                                           CF003110
       MDX     CLOOP           DO NEXT MULT                    CF003120
       SRT     16              PUT NUMBER IN 0                 CF003130
*              CONVERSION OF CHARACTER TO INTEGER              CF003140
*              DONE                                            CF003150
C1SV   LDX   L1 *-*             RESTORE XRS                     CF003160
C2SV   LDX   L2 *-*                                            CF003170
       BSI     1 FLOAT-R1PNT   CONVERT AND PUT IN              CF003180
       BSI     2 SWAP-R2PNT    IN LEFT                         CF003190
       LD      3 POINT-R3PNT   POWER OF TEN BY                 CF003200
*                              WHICH TO MULT                   CF003210
       BSC   L PGEZ,-          BRANCH ON ZERO OR +             CF003220
       EOR     2 HFFFF-R2PNT                                  CF003230
       A       2 K1-R2PNT                                     CF003240
       STO     3 POINT-R3PNT   NEGATIVE OF POINT               CF003250
       LD      MADR            POINT TO NEGATIVE POWER         CF003260
       MDX     *+1                                            CF003270
PGEZ   LD      PADR            POINT TO POSITIVE               CF003280
       STO     TABPT                                          CF003290
       LD      3 POINT-R3PNT                                  CF003300
PTEST  BSC   L MLTOV,+-        BRANCH ON ZERO                  CF003310
       BSC     E                                              CF003320
       MDX     *+1                                            CF003330
       MDX     LBOFF           BRANCH ON BIT OFF               CF003340
TABPT  EQU     *+1                                            CF003350
       LD    L *-*                                            CF003360
       BSI     3 FLDR-R3PNT    LOAD INTO RIGHT                 CF003370
       BSI     2 FMPY-R2PNT                                   CF003380
       BSI     2 SWAP-R2PNT                                   CF003390
LBOFF  MDX   L TABPT,2         BUMP TO NEXT POWER              CF003400
       LD      3 POINT-R3PNT                                  CF003410
       SRA     1                                              CF003420
       STO     3 POINT-R3PNT                                  CF003430
       MDX     PTEST                                          CF003440
MLTOV  BSI     2 SWAP-R2PNT                                   CF003450
       LD      3 MFLAG-R3PNT   GET SIGN FLAG                   CF003460
       BSC   L NCHNG,+-        BRANCH IF OFF                   CF003470
       BSI     2 SCHR-R2PNT    ON -- CHANGE SIGN               CF003480
       SRA     16              RESET                           CF003490
       STO     3 MFLAG-R3PNT   SIGN FLAG                       CF003500
NCHNG  EQU     *                                              CF003510
       BSI     1 PACKR-R1PNT  RSLT IN AC + Q                   CF003520
       BSC   I CONV             RETURN                        CF003530
PTABP  DEC     1.0E1                                          CF003540
       DEC     1.0E2                                          CF003550
       DEC     1.0E4                                          CF003560
```

```
        DEC     1.0E8                                                   CF003400
        DEC     1.0E16                                                  CF003410
        DEC     1.0E32                                                  CF003420
PTABM   DEC     1.0E-1                                                  CF003430
        DEC     1.0E-2                                                  CF003440
        DEC     1.0E-4                                                  CF003450
        DEC     1.0E-8                                                  CF003460
        DEC     1.0E-16                                                 CF003470
        DEC     1.0E-32                                                 CF003480
C10     DC      10                                                      CF003490
MADR    DC      PTABM                                                   CF003500
PADR    DC      PTABP                                                   CF003510
*                                                                       CF003520
*                                                                       CF003530
*       GTOPR                                                           CF003540
*               GETS NEXT OPERATOR                                      CF003550
*               RETURNS -1 IF NONE LEFT                                 CF003560
GTOPR   DC      0                                                       CF003570
        MDX  L  NMOPL,0                                                 CF003580
        MDX     SMOPL                                                   CF003590
        LD    2 HFFFF-R2PNT                                             CF003600
GGOBK   BSC  I  GTOPR                                                   CF003610
OPPLC   EQU     *+1                                                     CF003620
SMOPL   LD   L  *-*                                                     CF003630
        MDX  L  OPPLC,1                                                 CF003640
        BSC  L  SMOPL,+-                                                CF003650
        MDX  L  NMOPL,-1                                                CF003660
        MDX     GGOBK                                                   CF003670
        MDX     GGOBK                                                   CF003680
*                                                                       CF003690
*                                                                       CF003700
*       ORTOR                                                           CF003710
ORTOR   DC      0                                                       CF003720
        OR    3 LEVEL-R3PNT                                             CF003730
OSTKR   EQU     *+1                                                     CF003740
        STO  L  *-*                                                     CF003750
        MDX  L  OSTKR,1                                                 CF003760
        LD    2 HFFFF-R2PNT                                             CF003770
        STO  I  OSTKR           PUT -1 IN NEXT OPERATOR                 CF003780
*               POSITION                                                CF003790
        BSC  I  ORTOR                                                   CF003800
OSTAK   BSS     25                                                      CF003810
RSTAK   DEC     0                                                       CF003820
        BSS     50                                                      CF003830
RSEND   DC      RSTAK+50                                                CF003840
*                                                                       CF003850
*ORAND                                                                  CF003860
*                                                                       CF003870
ORAND   DC      0                                                       CF003880
RSTKR   EQU     *+1                                                     CF003890
        STD  L  *-*                                                     CF003900
        LD      RSEND                                                   CF003910
        S       RSTKR                                                   CF003920
        BSC  L  OERR,+                                                  CF003930
        MDX  L  RSTKR,2                                                 CF003940
        BSC  I  ORAND                                                   CF003950
OERR    BSI  L  ERROR                                                   CF003960
        DC      19                                                      CF003970
*                                                                       CF003980
*                                                                       CF003990
*               PTNST                                                   CF004000
*                                                                       CF004010
PTNST   DC      0                                                       CF004020
```

```
        LD      NSTND                                          CF004030
        S       NSTKR                                          CF004040
        BSC  L  NPUT,Z-         BRANCH IF OK >-+               CF004050
        BSI  L  ERROR           NEST OVERFLOW                  CF004060
        DC      20                                             CF004070
NPUT    LD    3 LEVEL-R3PNT     PUT IN NEST                    CF004080
NSTKR   EQU     *+1                                            CF004090
        STO  L  *-*                                            CF004100
        MDX  L  NSTKR,1         POINT TO NEXT ENTRY            CF004110
        BSC  I  PTNST                                          CF004120
NEST    BSS     4                                              CF004130
NSTND   DC      NEST+4                                         CF004140
NSTBG   DC      NEST                                           CF004150
*                                                              CF004160
*                                                              CF004170
*               GTNST                                          CF004180
*                                                              CF004190
*                                                              CF004200
GTNST   DC      0                                              CF004210
        LD      NSTKR                                          CF004220
        S       NSTBG                                          CF004230
        BSC  L  NGET,+          BR IN NEST EMPTY               CF004240
        MDX  L  NSTKR,-1  POINT TO LAST ENTRY                  CF004250
        LD   I  NSTKR                                          CF004260
        MDX  L  NSTKR,1  RETURN TO NEXT ENTRY                  CF004270
        MDX     *+1                                            CF004280
NGET    LD    2 HFFFF-R2PNT     LOAD -1                        CF004290
        BSC  I  GTNST                                          CF004300
*                                                              CF004310
*                                                              CF004320
*               BKNST                                          CF004330
*                                                              CF004340
BKNST.  DC      0                                              CF004350
        MDX  L  NSTKR,-1   BACK UP NEST POINTER                CF004360
        BSC  I  BKNST                                          CF004370
*                                                              CF004380
*                                                              CF004390
*                                                              CF004400
*               CMPRE                                          CF004410
*                                                              CF004420
*                                                              CF004430
AUTHR   DEC     0               AUTHOR RESULT                  CF004440
FUDGE   DEC     .01  ACCEPTABLE RELATIVE ERROR                 CF004450
*                                                              CF004460
*               ABS>R1 - R2+ LE FUDGE * R2                     CF004470
*               IS CONSIDERED MATCH                            CF004480
CMPRE   DC      0                                              CF004490
        STD     AUTHR           STORE AUTHORS RESULT           CF004500
        BSI   3 FLDR-R3PNT AUTHOR IN RIGHT                     CF004510
        LDD   3 RSLT1-R3PNT                                    CF004520
        BSI   3 FLDL-R3PNT  STUDENT  IN LEFT                   CF004530
        BSI   2 FSUB-R2PNT       UBTRACT                       CF004540
        BSI   1 FABS-R1PNT  ABSOLUTE VALUE                     CF004550
        BSI   1 PACKR-R1PNT  2 WORD IN AC                      CF004560
        STD   3 RSLT1-R3PNT     SAVE                           CF004570
        LDD     FUDGE      FUDGE FACTOR TO LEFT                CF004580
        BSI   3 FLDL-R3PNT                                     CF004590
        LDD     AUTHR                                          CF004600
        BSI   3 FLDR-R3PNT      AUTHOR TO RIGHT                CF004610
        BSI   2 FMPY-R2PNT      MULTIPLY                       CF004620
        BSI   1 FABS-R1PNT       ABS OF PRODUCT                CF004630
        LDD   3 RSLT1-R3PNT                                    CF004640
```

```
        BSI    3 FLDL-R3PNT      ABS OF DIFFERENCE IN
*                                LEFT
        BSI    2 FSUB-R2PNT
        BSI    1 PACKR-R1PNT     2 WORD IN AC + Q
        BSC  L   CMPOK,+
        LD     2 HFFFF-R2PNT     LOAD -1
CMRTN   STO  L   ATR23           IN COUNTER 23
        STO  L   ATR22
        BSC  I   CMPRE
CMPOK   SRA      16                LOAD ZERO
        MDX      CMRTN
*
*               TRVRS
*               GOES THROUGH STACKS
*               ANSWER IN AC + Q
*
TRVRS   DC       0
        SLT      32                ZERO AC AND Q
        LD   L   OSTKR             ADDRESS OF LAST OPERATO
*                                  R+1
        S        OPLOC             ADDRESS OF STACK BEGINN
        STO      NUMOP
        BSC  L   OGT1,Z
        LDD  L   RSTAK             NO OPERATORS LOAD
*                                  FIRST OPERAND
        BSC  I   TRVRS             PICK UP FIRST OPERAND
*                                  AND RETURN
*
OGT1    STO    3 NMOPL-R3PNT       NM OPERATORS LEFT TO
*               ACCESS
        BSI  L   GTOPR             GET OPERATOR
        STO      SVEOP             SAVE OPERATOR
GTAGN   BSI  L   GTOPR             GET NEXT
        BSC  L   TCALC,+           BRANCH IF -1 RETURNED
*               PRESENT OPERATOR IS IN THE AC
        STO    2 STAT-R2PNT        SAVE IN STAT
        AND    2 H000F-R2PNT       DELETE LEVEL
        S      2 K2-R2PNT          SEE IF 1 OR 2
        BSC  L   N1OR2,Z-          BRANCH IF NO
        LD       SVEOP             GET LAST OP
        AND    2 H000F-R2PNT       DELETE LEVEL
        S      2 K2-R2PNT          SEE IF 1 OR 2
        BSC  L   N1OR2,Z-          BRANCH IF NO
        LD     2 STAT-R2PNT        LOAD PRESENT OPERATOR
        OR     2 H000F-R2PNT       CLOBBER OP FIELD
        STO    2 FSUB-R2PNT        STORE IN FSUB
        LD       SVEOP             GET LAST OPERATOR
        OR     2 H000F-R2PNT       CLOBBER OPERATOR FIELD
        S      2 FSUB-R2PNT        SUBTRACT LAST OPERATOR
        BSC  L   TCALC,+-          BRANCH IF EQUAL
N1OR2   LD     2 STAT-R2PNT        RELOAD PRESENT
        S        SVEOP             SUBTRACT LAST
        BSC      +                 SKIP IF LAST LT PRESENT
        MDX      TCALC             CALCULATE BRANCH
        A        SVEOP             RESTORE PRESENT OP
        MDX      GTAGN-1           GET ANOTHER
TCALC   BSI      CALC
        MDX  L   NUMOP,0
        MDX      *+1
        MDX      OGT1-2            RETURN
```

```
*                ANSWER IN AC AND Q                                    CF005060
         LD      NUMOP                                                  CF005070
         MDX     OGT1                                                   CF005080
*                                                                       CF005090
*                CALC                                                   CF005100
*                                                                       CF005110
NUMOP    BSS     1               NUMBER OF OPERATORS IN                 CF005120
*                                STACK                                  CF005130
SVEOP    BSS     1               LAST OPERATOR ACCESSED                 CF005140
RPICK    DC      *-*             ADDR OF OPERAND TO BE                  CF005150
*                                PICKED UP                              CF005160
RNLOC    DC      RSTAK           OPERAND STACK BEGINNING                CF005170
OPLOC    DC      OSTAK           OPERATOR STACK BEGINNING               CF005180
*                                                                       CF005190
*                                                                       CF005200
*                                                                       CF005210
CALC     DC      0                                                      CF005220
         LD      OPLOC                                                  CF005230
         STO     OPICK           NEXT OPERATOR TO PICK                  CF005240
*                                FOR CALC                               CF005250
         STO  L  OPPLC           REINITIALIZE GTOPR                     CF005260
         LD      RNLOC                                                  CF005270
         STO     RPICK           NEXT OPERAND TO BE                     CF005280
*                                PICKED UP                              CF005290
         MDX  L  NUMOP,-1        DECREMENT NUMBER OF                    CF005300
*                                OPERATORS LEFT                         CF005310
         MDX     *                                                      CF005320
OPICK    EQU     *+1                                                    CF005330
         LD   L  *-*             GET OPERATOR                           CF005340
         EOR     SVEOP           COMPARE WITH SVEOP                     CF005350
         BSC  L  OPERF,+-        BRANCH IF EQUAL                        CF005360
         MDX  L  OPICK,1         POINT TO NEXT OPERATOR                 CF005370
         MDX  L  RPICK,2         AND ITS ASSOCIATED LEFT                CF005380
*                                OPERAND                                CF005390
         MDX     OPICK-1         RETURN FOR NEXT                        CF005400
OPERF    LDD  I  RPICK           GET LEFT OPERAND                       CF005410
         BSI     3 FLDL-R3PNT PUT IN LEFT                               CF005430
         MDX  L  RPICK,2         POINT TO NEXT OPERAND                  CF005440
         LD      OPICK           OPERAND ADDRESS                        CF005450
         A       2 K1-R2PNT      ADD 1 TO POINT TO NEXT                 CF005460
         STO     N2PIK           OPERATOR LOCATION                      CF005470
N2PIK    EQU     *+1                                                    CF005480
         LD   L  *-*             NEXT OPERATOR                          CF005490
         BSC  L  GOTND,Z         IN NON ZERO BRANCH                     CF005500
         MDX  L  N2PIK,1                                                CF005510
         MDX  L  RPICK,2                                                CF005520
         MDX     N2PIK-1                                                CF005530
*                                                                       CF005540
*                                                                       CF005550
GOTND    LDD  I  RPICK           RIGHT OPERAND                          CF005560
         BSI     3 FLDR-R3PNT    PUT IN RIGHT                           CF005570
         LD   I  OPICK           GET OPERATOR                           CF005580
         AND     2 H000F-R2PNT   DELETE LEVEL                           CF005590
         A       FTBL            ADDRESS OF ROUTINE                     CF005600
*                                TABLE                                  CF005610
         STO     CDOIT                                                  CF005620
         SRA     16                                                     CF005630
         STO  I  OPICK           ZERO OUT OPERATOR                      CF005640
CDOIT    EQU     *+1             GO TO FLOATING ROUTINE                 CF005650
         BSI  I  *-*                                                    CF005660
         BSI     1 PACKR-R1PNT ANSWER IN SECOND OPERAND PC              CF005670
```

```
*                                        POSITION
      STD   I  RPICK
      BSC   I  CALC
*
FTBL  DC       *-1
      DC       FSUB
      DC       FADD
      DC       FMPY
      DC       FDIV
      DC       FAXB
      DC       FLOG
*
*
*
*              BLD14
*
*              CONVERTS ALPHABETIC CHARACTER 0
*              TO NUMERIC 0
*
*
BLD14 LD     3 C0-R3PNT
*
*
*
*              BLD10
*                                  TAKES CARE OF NUMERIC
*
BLD10 STO    2 STAT-R2PNT
      SRA      16                  ZERO FLAGS
      STO      CRHLD
      STO      CRHLD+1
      STO      CRHLD+2
      STO      CRHLD+3
      STO    3 NMDIG-R3PNT
      STO      NMSIG
      STO    3 POINT-R3PNT
      LD     2 STAT-R2PNT
      BSC  L   B1001,+-            BRANCH ON DECIMAL
*                                  POINT ENTRY
B1008 BSI      GTNUM
      BSC  L   B1002,Z+            NON-NUMERIC CHARACTER
      EOR    3 DECPT-R3PNT
      BSC  L   B1003,+-            BRANCH IF DECIMAL PNT
      EOR    3 DECPT-R3PNT
      STO    2 STAT-R2PNT
      BSC  L   B1004,+-
      LD       NMSIG
      S      2 K4-R2PNT
      BSC  L   B1005,-             ERROR IF GE 4
      A      2 K4-R2PNT
      A        CRADR               COMPUTE ADDRESS
      STO      *+2
      LD     2 STAT-R2PNT
      STO  L   *-*
B1006 LD       NMSIG
      A      2 K1-R2PNT
      STO      NMSIG
B1011 LD     3 NMDIG-R3PNT
      A      2 K1-R2PNT
      STO    3 NMDIG-R3PNT
B1007 BSI  L   GTNXT
```

```
        MDX   L  NFLG,0          TEST FOR ALGEBRAIC      CF006050
        MDX      B1099                                   CF006055
        MDX      B1008                                   CF006060
B1099   SLA      16                                      CF006065
        STO   3  NFLG-R3PNT                              CF006070
        MDX      B1002           EXIT IF ON              CF006075
B1001   LD    3  NMDIG-R3PNT                            CF006080
        A     2  K1-R2PNT                               CF006085
        STO   3  POINT-R3PNT                            CF006090
        MDX      B1007                                   CF006095
B1002   BSI   L  BAKUP                                   CF006100
        MDX   L  POINT,0          SEE IF DECIMAL        CF006120
        MDX      *+1              POINT FOUND           CF006130
        MDX      B1013            BRANCH IF NO          CF006140
        MDX   L  NMDIG,0          NOW SEE IF ANY        CF006150
        MDX      B1013            DIGITS FOUND          CF006160
        BSI   L  ERROR            NO -- THIS IS         CF006170
        DC       22               ERROR 22 -- DECIMAL   CF006180
B1013   EQU      *                POINT BUT NO DIGITS   CF006190
        LD       NMSIG                                   CF006200
        STO   2  STAT-R2PNT                             CF006210
        S     2  K4-R2PNT                               CF006220
        BSC      Z-                                      CF006230
        SRA      16                                      CF006250
        A     2  K4-R2PNT                               CF006260
        STO      NMSIG                                   CF006270
        LD    3  POINT-R3PNT                            CF006280
        BSC   L  B1012,+-         BRANCH IF POINT ZERO  CF006290
        A     2  STAT-R2PNT        CALCULATE POINT       CF006300
        S        NMSIG                                   CF006310
        S     3  NMDIG-R3PNT                            CF006320
        S     2  K1-R2PNT                               CF006330
        STO   3  POINT-R3PNT                            CF006340
B1009   BSI   L  CONV                                    CF006350
        BSI   L  ORAND                                  CF006360
        LD    3  PREC-R3PNT                             CF006370
        BSC   L  B1010,+                                CF006380
        LD    2  K3-R2PNT         IMPLIED MULT          CF006390
        BSI   L  ORTOR                                  CF006400
B1010   LD    2  K1-R2PNT                               CF006410
        STO   3  PREC-R3PNT                             CF006420
        BSC   L  A1               RETURN                CF006430
B1003   LD    3  POINT-R3PNT                            CF006440
        BSC   L  B1001,+-         BRANCH ON POINT ZERO  CF006450
        BSI   L  ERROR                                  CF006460
        DC       13               MORE THAN 1 DECIMAL PT CF006470
B1004   LD       NMSIG                                   CF006480
        BSC   L  B1006,Z                                CF006490
        LD    3  POINT-R3PNT                            CF006500
        BSC   L  B1007,+-                               CF006520
        MDX      B1011                                   CF006530
B1005   BSI   L  ERROR            EXCESS SIGNIFICANT DGTSCF006540
        DC       14                                     CF006550
B1012   LD    3  NMDIG-R3PNT      NUM DIGITS AFTER 1ST NZCF006560
        S     2  K4-R2PNT                               CF006570
        BSC      Z-                                      CF006580
        STO   3  POINT-R3PNT      DIFFERENCE TO POINT   CF006590
        MDX      B1009                                   CF006600
CRADR   DC       CRHLD                                   CF006610
CRHLD   BSS      4                                       CF006620
NMSIG   EQU      CALC             NUMBER OF SIGNIFICANT  CF006630
*                                 DIGITS                CF006640
```

```
*                                                                    CF00664
*                                                                    CF00667
*           GTNUM                                                    CF00668
*           GETS NUMERIC OR DECIMAL CHARACTER                        CF00669
*           RETURNS BINARY NUMBER FOR NUMERIC                        CF00670
*           CHARACTER, THE DECIMAL POINT, OR                         CF00671
*           -1 IF CHARACTER IS NOT NUMERIC OR .                      CF00672
*                                                                    CF00673
GTNUM DC    0                                                        CF00674
      S     3 C0-R3PNT        SUBTRACT EBCDIC ZERO                   CF00675
      BSC   I GTNUM,-         RETURN WITH BINARY NUMB                CF00676
      A     3 C0-R3PNT        RESTORE                                CF00677
      EOR   3 DECPT-R3PNT                                            CF00678
      BSC   L LDDEC,+-                                               CF00679
      EOR   3 DECPT-R3PNT                                            CF00680
      AND   3 LMASK-R3PNT     CONVERT TO LOWER CASE                  CF00681
      EOR   3 LITL-R3PNT      TEST FOR L                             CF00682
      BSC   L ONELD,+-        BRANCH IF YES                          CF00683
      EOR   3 LITL-R3PNT                                             CF00684
      EOR   3 LITO-R3PNT                                             CF00685
      BSC   L ZROLD,+-        BRANCH IF 0                            CF00686
GNONE LD    2 HFFFF-R2PNT     NONE OF ABOVE, LOAD -1                 CF00687
      MDX     GTRTN                                                  CF00688
ONELD BSI   L GTNXT           CHECK NEXT                             CF00689
      AND   3 LMASK-R3PNT     CHARACTER FOR 0                        CF00690
      EOR   3 LITO-R3PNT                                             CF00691
      BSC   L GT001,Z         BRANCH IF NOT                          CF00692
      BSI   L GTNXT           CHECK NEXT                             CF00693
      AND   3 LMASK-R3PNT     CHARACTER FOR G                        CF00694
      EOR     SMALG                                                  CF00695
      BSC   L GT002,+-        BRANCH IF IT IS                        CF00696
      BSI   L BAKUP                                                  CF00697
GT001 BSI   L BAKUP                                                  CF00698
      LD    2 K1-R2PNT        PICK UP 1                              CF00699
      MDX     GTRTN           RETURN                                 CF00700
GT002 BSI   L BAKUP                                                  CF00701
      BSI   L BAKUP                                                  CF00702
      MDX     GNONE                                                  CF00703
ZROLD SLA     16                                                     CF00704
GTRTN BSC   I GTNUM                                                  CF00705
LDDEC LD    3 DECPT-R3PNT                                            CF00706
      MDX     GTRTN                                                  CF00707
*                                                                    CF00708
*                                                                    CF00709
*           BLD05                                                    CF00710
*           TAKES CARE OF LOG BASE 10                                CF00711
*                                                                    CF00712
BLD05 BSI   L GTNXT                                                  CF00713
      AND   3 LMASK-R3PNT     CONVERT TO LOWER CASE                  CF00714
      EOR   3 LITO-R3PNT                                             CF00715
      BSC   L LOGO,+-                                                CF00716
      MDX     BAK05                                                  CF00717
BK051 BSI   L BAKUP                                                  CF00718
BAK05 BSI   L BAKUP           NUMERIC -- BACKUP                      CF00719
      LD      C1              PUT F1 IN AC                           CF00720
      BSC   L BLD10           GO TO NUMERIC                          CF00721
C1    DC      /00F1                                                  CF00722
SMALG DC      /0087                                                  CF00723
LOGO  BSI   L GTNXT           CHECK FOR G                            CF00724
      AND   3 LMASK-R3PNT                                            CF00725
      EOR     SMALG                                                  CF00726
```

```
         BSC      Z                                                  CF007210
         MDX      BK051                                              CF007220
         LD     3 PREC-R3PNT       CHECK PRECEEDING                 CF007230
         BSC  L   RLAST,-Z                                          CF007240
LEVIN    LD       H0020            INCREMENT LEVEL                  CF007250
         A      3 LEVEL-R3PNT                                       CF007260
         STO    3 LEVEL-R3PNT                                       CF007270
         LD     2 K6-R2PNT         LOG OPERATOR                     CF007280
         BSI  L   ORTOR                                             CF007290
         BSI  L   OPAND            DUMMY OPERAND                    CF007300
CKLPR    BSI  L   GTNXT                                             CF007310
         EOR  L   LPREN            CHECK FOR LEFT PAREN             CF007320
         BSC  L   LPOK,+-          BRANCH IF FOUND                  CF007330
         EOR  L   LPREN            NOT FOUND                        CF007340
         EOR  L   DOWN             CHECK FOR LF                     CF007350
         BSC  L   SUBSC,+-         LOG BASE GIVEN                   CF007360
LV15     BSI  L   ERROR            NO LEFT PAREN                    CF007370
         DC       15                                                CF007380
RLAST    LD     2 K3-R2PNT         OPERAND LAST -- LOAD             CF007390
         BSI  L   ORTOR            IMPLIED MULTIPLIER               CF007400
         MDX      LEVIN            BUMP LEVEL                       CF007410
H0020    DC       /0020                                             CF007420
LPOK     BSI  L   PTNST            LEFT PAREN FOUND                 CF007430
         BSI  L   BAKUP            STORE LEVEL AND BACKUP           CF007440
         SRA      16                                                CF007450
         STO    3 PREC-R3PNT       ZERO PREC                        CF007460
         BSC  L   A1                                                CF007470
SUBSC    BSI  L   GTNXT            LINE FEED FOUND                  CF007480
         EOR  L   UP               EGN ALL UNTIL NEXT UP            CF007490
*                                  FOUND                            CF007500
         BSC  L   CKLPR,+-                                          CF007510
         EOR  L   UP               NOT FOUND                        CF007520
         EOR  L   EOBB             IS IT EOB                        CF007530
         BSC  L   SUBSC,Z          NO GET NEXT                      CF007540
         MDX      LV15             YES -- ERROR                     CF007550
*                                                                   CF007560
*                                                                   CF007570
*              BLD06                                                CF007580
*              LEFT PARENTHESIS                                     CF007590
*                                                                   CF007600
BLD06    LD     3 PREC-R3PNT                                        CF007610
         BSC  L   B0601,Z-          BRANCH IF OPERAND LAST           CF007620
B0602    LD     3 H0030-R3PNT       BUMP LEVEL                       CF007630
         A      3 LEVEL-R3PNT                                       CF007640
         STO    3 LEVEL-R3PNT                                       CF007650
         LD     2 HFFFF-R2PNT       PUT -1 IN PREC                  CF007660
         STO    3 PREC-R3PNT                                        CF007670
         BSC  L   A1                                                CF007680
B0601    LD     2 K3-R2PNT          LOAD IMPLIED MULT               CF007690
         BSI  L   ORTOR                                             CF007700
         MDX      B0602                                             CF007710
*                                                                   CF007720
*                                                                   CF007730
*              BLD07                                                CF007740
*              RIGHT PARENTHESIS                                    CF007750
*                                                                   CF007760
BLD07    LD     3 PREC-R3PNT                                        CF007770
         BSC  L   B0701,Z                                           CF007780
         BSI  L   ERROR             OPERATOR ENDS EXPRES            CF007790
         DC       10                                                CF007800
B0701    BSC  L   B0702,-                                           CF007810
```

```
        BSC      Z                                                  CF007210
        MDX      BK051                                              CF007220
        LD     3 PREC-R3PNT       CHECK PRECEEDING                 CF007230
        BSC  L   RLAST,-Z                                          CF007240
LEVIN   LD       H0020            INCREMENT LEVEL                  CF007250
        A      3 LEVEL-R3PNT                                       CF007260
        STO    3 LEVEL-R3PNT                                       CF007270
        LD     2 K6-R2PNT         LOG OPERATOR                     CF007280
        BSI  L   ORTOR                                             CF007290
        BSI  L   ORAND            DUMMY OPERAND                    CF007300
CKLPR   BSI  L   GTNXT                                             CF007310
        EOR  L   LPREN            CHECK FOR LEFT PAREN             CF007320
        BSC  L   LPOK,+-          BRANCH IF FOUND                  CF007330
        EOR  L   LPREN            NOT FOUND                        CF007340
        EOR  L   DOWN             CHECK FOR LF                     CF007350
        BSC  L   SUBSC,+-         LOG BASE GIVEN                   CF007360
LV15    BSI  L   ERROR            NO LEFT PAREN                    CF007370
        DC       15                                                CF007380
RLAST   LD     2 K3-R2PNT         OPERAND LAST -- LOAD             CF007390
        BSI  L   ORTOR            IMPLIED MULTIPLIER               CF007400
        MDX      LEVIN            BUMP LEVEL                       CF007410
H0020   DC       /0020                                             CF007420
LPOK    BSI  L   PTNST            LEFT PAREN FOUND                 CF007430
        BSI  L   BAKUP            STORE LEVEL AND BACKUP           CF007440
        SRA      16                                                CF007450
        STO    3 PREC-R3PNT       ZERO PREC                        CF007460
        BSC  L   A1                                                CF007470
SUBSC   BSI  L   GTNXT            LINE FEED FOUND                  CF007480
        EOR  L   UP               EGN ALL UNTIL NEXT UP            CF007490
*                                 FOUND                            CF007500
        BSC  L   CKLPR,+-                                          CF007510
        EOR  L   UP               NOT FOUND                        CF007520
        EOR  L   EOBB             IS IT EOB                        CF007530
        BSC  L   SUBSC,Z          NO GET NEXT                      CF007540
        MDX      LV15             YES -- ERROR                     CF007550
*                                                                  CF007560
*                                                                  CF007570
*           BLD06                                                  CF007580
*           LEFT PARENTHESIS                                       CF007590
*                                                                  CF007600
BLD06   LD     3 PREC-R3PNT                                        CF007610
        BSC  L   B0601,Z-         BRANCH IF OPERAND LAST           CF007620
B0602   LD     3 H0030-R3PNT      BUMP LEVEL                       CF007630
        A      3 LEVEL-R3PNT                                       CF007640
        STO    3 LEVEL-R3PNT                                       CF007650
        LD     2 HFFFF-R2PNT      PUT -1 IN PREC                   CF007660
        STO    3 PREC-R3PNT                                        CF007670
        BSC  L   A1                                                CF007680
B0601   LD     2 K3-R2PNT         LOAD IMPLIED MULT                CF007690
        BSI  L   ORTOR                                             CF007700
        MDX      B0602                                             CF007710
*                                                                  CF007720
*                                                                  CF007730
*           BLD07                                                  CF007740
*           RIGHT PARENTHESIS                                      CF007750
*                                                                  CF007760
BLD07   LD     3 PREC-R3PNT                                        CF007770
        BSC  L   B0701,Z                                           CF007780
        BSI  L   ERROR            OPERATOR ENDS EXPRES             CF007790
        DC       10                                                CF007800
B0701   BSC  L   B0702,-                                           CF007810
```

```
         BSI  L  ERROR                                          CF007820
         DC      11                                             CF007830
B0702    LD     3 LEVEL-R3PNT     DECREMENT LEVEL              CF007840
         S      3 H0030-R3PNT                                  CF007850
         STO    3 LEVEL-R3PNT                                  CF007860
         BSI  L  GTNST             GET LAST STORED LEVEL        CF007870
         S      3 LEVEL-R3PNT                                  CF007880
         BSC  L  B0703,Z           BRANCH IF NOT EQUAL          CF007890
         LD     3 LEVEL-R3PNT     DECREMENT LEVEL AGAIN        CF007900
         S       H0020                                         CF007910
         STO    3 LEVEL-R3PNT                                  CF007920
         BSI  L  BKNST             BACKUP NEST                  CF007930
B0703    LD     3 LEVEL-R3PNT                                  CF007940
         BSC  L  A1,-                                          CF007950
         BSI  L  ERROR             LEVEL NEGATIVE               CF007960
         DC      12                                            CF007970
*                                                              CF007980
*                                                              CF007990
*                                                              CF008000
*              BLD08                                           CF008010
*              POWER OPERATOR                                  CF008020
*                                                              CF008030
BLD08    LD     3 PREC-R3PNT                                   CF008040
         BSC  L  B0801,Z                                       CF008050
         BSI  L  ERROR                                         CF008060
         DC      7                                             CF008070
B0801    BSC  L  B0802,-                                       CF008080
         BSI  L  ERROR                                         CF008090
         DC      8                                             CF008100
B0802    LD     2 K5-R2PNT                                     CF008110
         BSI  L  ORTOR                                         CF008120
         LD     3 LEVEL-R3PNT     INCREMENT LEVEL AFTER        CF008130
         A       H0010            INSERTING OPERATOR           CF008140
         STO    3 LEVEL-R3PNT                                  CF008150
         LD     2 HFFFF-R2PNT                                  CF008160
         STO    3 PREC-R3PNT                                   CF008170
         BSC  L  A1                                            CF008180
H0010    DC      /0010                                         CF008190
*                                                              CF008200
*                                                              CF008210
*              BLD09                                           CF008220
*              POWER EXPRESSION END                            CF008230
*                                                              CF008240
BLD09    LD     3 PREC-R3PNT                                   CF008250
         BSC  L  B0901,-                                       CF008260
         BSI  L  ERROR                                         CF008270
         DC      16                                            CF008280
B0901    BSC  L  B0902,Z                                       CF008290
         BSI  L  ERROR                                         CF008300
         DC      17                                            CF008310
B0902    LD     3 LEVEL-R3PNT                                  CF008320
         S       H0010                                         CF008330
         STO    3 LEVEL-R3PNT                                  CF008340
         BSC  L  A1,-                                          CF008350
         BSI  L  ERROR                                         CF008360
         DC      12                                            CF008370
*                                                              CF008380
*                                                              CF008390
*              BLD11                                           CF008400
*              EOB HANDLINE                                    CF008410
*                                                              CF008420
```

```
BLD11 LD     3 PREC-R3PNT                                          CF008430
      BSC  L   B1100,Z                                           CF008440
      BSI  L   ERROR                                             CF008450
      DC       10                                                CF008460
B1100 BSC  L   B1101,-                                           CF008470
      BSI  L   ERROR                                             CF008480
      DC       18                                                CF008490
B1101 LD     3 LEVEL-R3PNT                                       CF008500
      BSC  I   BUILD,+-                                          CF008510
      LD     3 LEVEL-R3PNT                                       CF008520
      S    L   H0010                                             CF008530
      BSC  I   BUILD,+-                                          CF008540
      BSI  L   ERROR                                             CF008550
      DC       12                                                CF008560
*                                                                CF008570
*                                                                CF008580
*              SET UP PROGRAM SWITCHES                           CF008590
*                                                                CF008600
SETUP DC       0                                                 CF008610
*              INITIALIZE GTNXT START                            CF008620
      STO  L   CURAD                                             CF008630
*              INITIALIZE ERROR                                  CF008640
      STO      BKSTR                                             CF008650
      A      3 K50-R3PNT                                         CF008660
*              INITIALIZE GTNXT END                              CF008670
      STO  L   SP50                                              CF008680
      SLT      32                                                CF008690
*              INITIALIZE GTNXT L OR R CHARACTER                 CF008700
      STO  L   LEFTC                                             CF008710
*              PRIORITY OF OPERATION                             CF008720
      STO    3 LEVEL-R3PNT                                       CF008730
*              INITIALIZE SIGN FLAG                              CF008731
      STO    3 MFLAG-R3PNT                                       CF008732
*              INITIALIZE POWER LEVEL FLAG                       CF008733
      STO    3 LINE-R3PNT                                        CF008734
      LD       NAD                                               CF008740
*              ADDRESS OF FUNCTION NEST STACK                    CF008750
      STO  L   NSTKR                                             CF008760
      LD       OAD                                               CF008770
*              INITIALIZE ORTOR                                  CF008780
      STO  L   OSTKR                                             CF008790
*              INITIALIZE GTOPR                                  CF008800
      STO  L   OPPLC                                             CF008810
      LD       RAD                                               CF008820
*              INITIALIZE ORAND                                  CF008830
      STO  L   RSTKR                                             CF008840
      LD     2 HFFFF-R2PNT                                       CF008850
*              SET PREC TO SHOW EXPRESSION BEGINNING             CF008860
      STO    3 PREC-R3PNT                                        CF008870
      BSC  I   SETUP                                             CF008880
OAD   DC       OSTAK                                             CF008890
NAD   DC       NEST                                              CF008900
RAD   DC       RSTAK                                             CF008910
*                                                                CF008920
*                                                                CF008930
*              ERROR                                             CF008940
*              HANDLES ANY KIND OF ERROR                         CF008950
*              PUTS CURRENT CHARACTER NUM IN                     CF008960
*              COUNTER 22                                        CF008970
*              PUTS ERROR CODE IN COUNTER 23                     CF008980
*                                                                CF008990
```

```
ERROR DC      0                                                    CF009050
      LD   L  CURAD          CURRENT ADDRESS                       CF009060
      S       BKSTR                                                CF009070
      SLA     1              MULTIPLY BY 2                         CF009080
      STO  L  ATR22                                                CF009090
      LD   I  ERROR                                                CF009100
      STO  L  ATR23                                                CF009110
      BSC  I  PROG                                                 CF009120
BKSTR DC      0                                                    CF009130
*                                                                  CF009140
*                                                                  CF009150
*                                                                  CF009160
*                                                                  CF009170
*                                                                  CF009180
*     FAXB                                                         CF009190
*     RAISES A TO THE B POWER                                      CF009200
*                                                                  CF009210
FAXB  DC      0                                                    CF009220
      LD     1 RIGHT-1-R1PNT                                       CF009230
      STO     FEND-1                                               CF009240
      LDD    1 RIGHT-R1PNT    SAVE RIGHT                           CF009250
      STD     FEND                                                 CF009260
      LD     1 LEFT-R1PNT     BASE FRACTION                        CF009270
      BSC  L  AGTZ,Z-         BRANCH IF A GT 0                     CF009280
      BSC  L  ALTZ,Z          BRANCH IF A LT 0                     CF009290
      LD     1 RIGHT-R1PNT                                         CF009300
      BSC  L  BGEZ,-          BRANCH IF A EQ 0 ,A. BGE0            CF009310
      BSI  L  ERROR                                                CF009320
      DC      4               0 ** -B                              CF009330
BGEZ  BSC  L  BGTZ,Z          A EQ 0 ,A. B GT 0                    CF009340
      BSI  L  ERROR                                                CF009350
      DC      5               0 ** 0                               CF009360
BGTZ  LDD    2 ZERO-R2PNT 0 ** +B                                  CF009370
FRTN1 BSI    3 FLDR-R3PNT                                          CF009380
FRTN2 BSC  I  FAXB                                                 CF009390
AGTZ  LD     1 RIGHT-R1PNT                                         CF009400
      BSC  L  BNEZ,Z          A GT 0 ,A. B NE Z                    CF009410
LDONE LDD    2 ONE-R2PNT      A ** 0                               CF009420
      MDX     FRTN1                                                CF009430
BNEZ  BSI     FDOIT           A ** B     ALL OK                    CF009440
      MDX     FRTN2                                                CF009450
ALTZ  LD     1 RIGHT-R1PNT                                         CF009460
      BSC     +-                                                   CF009470
      MDX     LDONE           -A ** 0                              CF009480
      LD     2 IFLG-R2PNT INTEGER EXPONENT FLAG                    CF009490
      BSC  L  INT,Z+                                               CF009500
      BSI  L  ERROR                                                CF009510
      DC      6               -A**B B NOT INTEGER                  CF009520
INT   BSI    1 FABS-R1PNT     ABS≥R+                               CF009530
      BSI    2 SWAP-R2PNT     B A                                  CF009540
      BSI    1 FABS-R1PNT     ABS≥A+                               CF009550
      BSI    2 SWAP-R2PNT     A B                                  CF009560
      BSI     FDOIT           COMPUTE -- ALL OK                    CF009570
      LD      FEND            ORIGINAL RIGHT                       CF009580
      BSC  L  BGTZ2,Z-        B GT 0                               CF009590
      LDD    2 ONE-R2PNT      COMPUTE                              CF009600
      BSI    3 FLDL-R3PNT     1 / RESULT                           CF009610
      BSI    2 FDIV-R2PNT                                          CF009620
BGTZ2 BSI    1 PACKR-R1PNT    SAVE ANSWER                          CF009630
      STD     FEND2           IN FEND2                             CF009640
      LD      FEND-1                                               CF009650
```

```
        STO   1 RIGHT-1-R1PNT  RESTORE OLD RIGHT
        LDD     FEND
        STO   1 RIGHT-R1PNT
        BSI   3 IFIX-R3PNT
        BSC  L  ODD,E
        LDD     FEND2          PUT ANSWER IN RIGHT
        MDX     FRTN1
ODD     LDD     FEND2          GET ANSWER
        BSI   3 FLDR-R3PNT     PUT IN RIGHT
        BSI   2 SCHR-R2PNT     CHANGE SIGN OF RIGHT
        MDX     FRTN2
FDOIT   DC      0
        BSI   2 SWAP-R2PNT     CHANGE B AND A
        BSI   3 FLN-R3PNT      LN≥A↑
        BSI   2 FMPY-R2PNT  B*LN≥A↑
        BSI   3 FXPN-R3PNT     EXP≥B*LN≥A↑↑
        BSC  I  FDOIT
        DC      0
FEND    DEC     0
FEND2   DEC     0
*
*
*
*               PACKR
*               STORES 3 WORD NUMBER FROM RIGHT INTO
*               RGT2
*
*
PACKR   DC      0
        STS   2 STAT-R2PNT     RESET OVDRFLOW
        BSI   2 SAVER-R2PNT    SAVE RIGHT
        LDD   1 RIGHT-R1PNT
        BSC     +Z             SKIP IF RIGHT POSITIVE
*                              OR ZERO
        BSI   2 SCHR-R2PNT
        AD    2 ROUND-R2PNT    %128 9/0000 0080
        BSC  L  POVFL,0        BRANCH IF OVERFLOW
PSTOR   STD   1 RIGHT-R1PNT    RETURN TO RIGHT
        LDD   1 RSAVE-R1PNT    CHECK IF RIGHT LT 0
        BSC     +Z
        BSI   2 SCHR-R2PNT CHANGE SIGN IF IT WAS
        LDD   1 RIGHT-R1PNT
        RTE     16             SWAP AC → Q
        AND   2 HFF00-R2PNT STRIP RIGHTMOST BITS
        OR    1 RIGHT-1-R1PNT
*               INSERT CHARACTERISTIC
        RTE     16             SWAP AGAIN
        BSC  I  PACKR
POVFL   LD    1 RIGHT-1-R1PNT  LOAD CHARACTERISTIC
        A     2 K1-R2PNT =%1
        STO   1 RIGHT-1-R1PNT
        BSI   2 FRNG-R2PNT
        LDD   2 H4000-R2PNT =9/4000 0000 MIN VALUE
        MDX     PSTOR
*
R1PNT   EQU     *
*
*
*
```

```
*              FABS                                                  CF01014
*              ABSOLUTE VALUE OF RIGHT                               CF01015
*                                                                    CF01020
*                                                                    CF01021
FABS  DC       0                                                     CF01022
      LD     1 RIGHT-R1PNT                                           CF01023
      BSC      Z+                                                    CF01024
      BSI    2 SCHR-R2PNT                                            CF01025
      BSC  I   FABS                                                  CF01026
*                                                                    CF01027
*                                                                    CF01028
*                                                                    CF01029
*              FLOAT                                                 CF01030
*              INTEGER IN A + Q                                      CF01031
*              FLOATING EQUIVALENT PLACED IN RIGHT                   CF01032
*                                                                    CF01033
FLOAT DC       0                                                     CF01034
      STO    1 RIGHT-R1PNT                                           CF01035
      LD     2 K159-R2PNT                                            CF01036
      STO    1 RIGHT-1-R1PNT                                         CF01037
      BSI    2 NORM-R2PNT                                            CF01038
      BSC  I   FLOAT                                                 CF01039
*                                                                    CF01040
*                                                                    CF01041
*                                                                    CF01042
*              CONSTANTS UNDER R1PNT                                 CF01043
*                                                                    CF01044
*                                                                    CF01045
QUOTE DEC      0                 ORDER DEPENDENT                     CF01047
      DEC      0                                                     CF01048
LEFT  DEC      0                                                     CF01050
      DEC      0                                                     CF01051
RIGHT DEC      0                                                     CF01052
      DEC      0                                                     CF01053
LSAVE DEC      0                                                     CF01054
      DEC      0                                                     CF01055
RSAVE DEC      0                                                     CF01056
QSAVE DEC      0                                                     CF01057
TSAVE DEC      0                                                     CF01058
J     DC       0                                                     CF01059
*                                                                    CF01060
*                                                                    CF01061
*                                                                    CF01062
*              FLOATING LOG BASE 10                                  CF01063
*              ARGUMENT IN RIGHT                                     CF01064
*              RESULT RETURNED TO RIGHT                              CF01065
*              LOG10≥X↑ ≠ LN≥X↑ / LN≥10↑                             CF01066
*                                                                    CF01067
*                                                                    CF01068
FLOG  DC       0                                                     CF01069
      BSI    3 FLN-R3PNT       TAKE NATURAL LOG                      CF01070
      BSI    2 SWAP-R2PNT      PUT RESULT IN LEFT                    CF01071
      LDD      BCON            GET LN≥10↑                            CF01072
      BSI    3 FLDR-R3PNT      PUT IN RIGHT                          CF01073
      BSI    2 FDIV-R2PNT      DIVIDE                                CF01074
      BSC  I   FLOG                                                  CF01075
BCON  DEC      2.3025851                                             CF01076
*                                                                    CF01077
*                                                                    CF01405
*                                                                    CF01079
*                                                                    CF01080
```

```
*
*
*
*
*      CONSTANTS UNDER R2PNT
*
*
ZERO . DEC      0
MPYER DEC       0
PROD  DEC       0
P5    DEC       0.5
ONE   DEC       1.0
TWO   DEC       2.0
ROUND DEC       128
H4000 DC        /4000           MUST BE EVEN
      DC        /0000
H7FFF DC        /7FFF           MUST BE EVEN
      DC        /FFFF
H8000 DC        /8000           MUST BE EVEN
      DC        /0000
K1    DC        1
K2    DC        2
K3    DC        3
K4    DC        4
K5    DC        5
K6    DC        6
K32   DC        32
K126  DC        126
K128  DC        128
K129  DC        129
K130  DC        130
K143  DC        143
K159  DC        159
H000F DC        /000F
H00FF DC        /00FF
IFLG  DC        0
HC000 DC        /C000
HFF00 DC        /FF00
HFFFF DC        /FFFF
STAT  DC
*
*
*
*             FADD AND FSUB
*             INPUT IN LEFT AND RIGHT
*             OUTPUT IN RIGHT
*
*
FSUB  DC        0
      BSI    2 SCHR-R2PNT      CHANGE SIGN OF RIGHT
      MDX       FAD1
FADD  DC        0
      LD        FADD           SET RETURN ADDRESS
      STO       FSUB
FAD1  BSI    2 SAVEL-R2PNT     SAVE LEFT ARG
*               SAVEL LEAVES EXP IN AC
      S      1 RIGHT-1-R1PNT
      BSC       Z-
      BSI    2 SWAP-R2PNT
      LD     1 RIGHT-1-R1PNT   LARGEST VALUE
      BSI    2 MOVEL-R2PNT
      STS    2 STAT-R2PNT RESET OVERFLOW INDICATOR
```

```
        AD    1 RIGHT-R1PNT                                          CF01142
        STD   1 RIGHT-R1PNT                                          CF01143
        BSC  L  OFLW,0          BRANCH IF OVERFLOW                   CF01144
OFT     BSI   3 LOADL-R3PNT     RESTORE LEFT ARGUMENT                CF01145
        BSI   2 NORM-R2PNT                                           CF01146
        BSI   2 FRNG-R2PNT                                           CF01147
        BSC  I  FSUB                                                 CF01148
*                                                                    CF01149
OFLW    SRT     1                                                    CF01150
        EOR   2 H8000-R2PNT     OVERFLOW CHANGES SIGN                CF01151
        STD   1 RIGHT-R1PNT     STORE                                CF01152
        LD    1 RIGHT-1-R1PNT   ADD TO EXP                           CF01153
        A     2 K1-R2PNT                                             CF01154
        STO   1 RIGHT-1-R1PNT                                        CF01155
        MDX     OFT                                                  CF01156
*                                                                    CF01157
*                                                                    CF01158
*                                                                    CF01159
*               SAVER AND SAVEL                                      CF01160
*               SAVES RIGHT AND LEFT RESPECTIVELY                    CF01161
*                                                                    CF01162
*                                                                    CF01163
SAVER   DC      0                                                    CF01164
        LDD   1 RIGHT-R1PNT                                          CF01165
        STD   1 RSAVE-R1PNT                                          CF01166
        LD    1 RIGHT-1-R1PNT                                        CF01167
        STO   1 RSAVE-1-R1PNT                                        CF01168
        BSC  I  SAVER                                                CF01169
*                                                                    CF01170
*                                                                    CF01171
SAVEL   DC      0                                                    CF01172
        MDX   1 -4                                                   CF01173
        BSI     SAVER                                                CF01174
        MDX   1 4                                                    CF01175
        BSC  I  SAVEL                                                CF01176
*                                                                    CF01177
*                                                                    CF01178
*                                                                    CF01179
*               FMPY                                                 CF01180
*               INPUT IN LEFT AND RIGHT                              CF01181
*               OUTPUT IN RIGHT                                      CF01182
*                                                                    CF01183
*                                                                    CF01184
FMPY    DC      0                                                    CF01185
        LD    1 RIGHT-1-R1PNT                                        CF01186
        A     1 LEFT-1-R1PNT                                         CF01187
        S     2 K128-R2PNT      REMOVE DOUBLE BIAS                   CF01188
        STO   1 RIGHT-1-R1PNT                                        CF01189
        LDD   1 LEFT-R1PNT                                           CF01190
        BSI   2 MPFRC-R2PNT                                          CF01191
        STD   1 RIGHT-R1PNT                                          CF01192
        BSI   2 NORM-R2PNT                                           CF01193
        BSI   2 FRNG-R2PNT                                           CF01194
        BSC  I  FMPY                                                 CF01195
*                                                                    CF01196
*                                                                    CF01197
*                                                                    CF01198
*               MPFRC                                                CF01199
*               MULTIPLIES FRACTIONS                                 CF01200
*                                                                    CF01201
*                                                                    CF01202
```

```
MPFRC DC      0                  A1 + A2 IN RIGHT
XMDS  EQU     MPFRC
*             B1 + B2 IN MPYER
*             A1*B1+A1*B2+A2*B1 PUT IN A AND Q
      STD   2 MPYER-R2PNT
      M     1 RIGHT-R1PNT
      STD   2 PROD-R2PNT       A1 * B1
      LD    2 MPYER+1-R2PNT
      SRA     1                CLEAR SIGN BIT
      M     1 RIGHT-R1PNT      A1 * B2
      SRT     15               ALIGN
      AD    2 PROD-R2PNT
      STD   2 PROD-R2PNT
      LD    1 RIGHT+1-R1PNT
      SRA     1                CLEAR SIGN BIT
      M     2 MPYER-R2PNT
      SRT     15
      AD    2 PROD-R2PNT
      SLT     1                REALIGN
      BSC   I MPFRC
*
*
*
*             SWAP
*             SWITCH LEFT AND RIGHT
*
*
SWAP  DC      0
      LDD   1 LEFT-R1PNT       SWAP FRACTIONS
      STD   1 TSAVE-R1PNT
      LDD   1 RIGHT-R1PNT
      STD   1 LEFT-R1PNT
      LDD   1 TSAVE-R1PNT
      STD   1 RIGHT-R1PNT
      LD    1 LEFT-1-R1PNT SWAP EXPONENTS
      SRT     16
      LD    1 RIGHT-1-R1PNT
      STO   1 LEFT-1-R1PNT
      SLT     16
      STO   1 RIGHT-1-R1PNT
      BSC   I SWAP
R2PNT EQU     *
*
*
*
*
*             MOVER AND MOVEL
*             SHIFT FRACTION AS REQUIRED
*             NO EXPONENT CHANGE
*
*
MOVER DC      0
*             ASSUME BIASED EXPONENT IN AC
      S     1 RIGHT-1-R1PNT
      BSC   L NGEQ,-           B IN N GE REXP THAT IS
*             SHIFT RIGHT REQUIRED
*             OTHERWISE LEFT REQUIRED
      A     2 K32-R2PNT
KGTT  BSC     *
SLT33 SLT     33
      EOR   2 HFFFF-R2PNT
```

```
         A        SLT33          /10A1                          CF01244
STORT    STO      BINPT                                         CF01245
         STS   2  XMDS-R2PNT                                    CF01246
         LDD   1  RIGHT-R1PNT                                   CF01247
BINPT    DC       *-*                                           CF01248
         BSC   I  MOVER                                         CF01249
NGEQ     S     2  K32-R2PNT                                     CF01270
KLTT     BSC      -                                             CF01271
SRT32    SRT      32                                            CF01272
         A        SRT32                                         CF01273
         MDX      STORT                                         CF01274
*                                                               CF01275
*                                                               CF01276
MOVEL    DC       0                                             CF01277
         MDX   1  -4                                            CF01278
         BSI      MOVER                                         CF01279
         MDX   1  4                                             CF01280
         BSC   I  MOVEL                                         CF01281
*                                                               CF01282
*                                                               CF01283
*                                                               CF01284
*              FRNG                                             CF01285
*              CHECK FOR OVER OR UNDER FLOW                     CF01286
*                                                               CF01287
*                                                               CF01288
FRNG     DC       0                                             CF01289
         LD    1  RIGHT-1-R1PNT                                 CF01290
         BSC   L  UNDER,Z+        B IF UNDERFLOW                CF01291
         SRA      8                                             CF01292
         BSC   I  FRNG,-+                                       CF01293
         BSI   L  ERROR                                         CF01294
         DC       1                                             CF01295
UNDER    BSI   L  ERROR                                         CF01296
         DC       2                                             CF01297
*                                                               CF01298
*                                                               CF01299
*                                                               CF01300
*              NORM                                             CF01301
*              NORMALIZE 3 WORD FLOATING POINT NUM              CF01302
*                                                               CF01303
*                                                               CF01304
NORM     DC       0                                             CF01305
         STS   2  STAT-R2PNT                                    CF01306
         SLT      32 RESET OVERFLOW AND ZERO AC                 CF01307
         SD    1  RIGHT-R1PNT                                   CF01308
         BSC      Z+              SKIP IF RIGHT LE 0            CF01309
*                 BUT NOT /8000 0000                            CF01310
         LDD   1  RIGHT-R1PNT                                   CF01311
         BSC   L  EXC,O           BRANCH IF RIGHT /80000        CF01312
         LDX   2  32              MAXIMUM ALIGNMENT             CF01313
         SLT      1               COUNT STOPS AT SHIFT TO       CF01314
*                 CARRY NOT SIGN                                CF01315
         SLC   2                  LEFT UNTIL CARRY HIT          CF01316
         STX   2  NT2                                           CF01317
         LD       NT2                                           CF01318
         LDX   L2 R2PNT           RESTORE XR2                   CF01319
         S     2  K32-R2PNT                                     CF01320
ADDQ     A     1  RIGHT-1-R1PNT                                 CF01321
         STO      NT2                                           CF01322
         BSI   2  MOVER-R2PNT     MOVE ACTUAL RIGHT ARG         CF01323
         STD   1  RIGHT-R1PNT     STORE IT                      CF01324
```

```
        BSC     Z               SKIP IF FRACTION IS ZERO
        LD      NT2             LOAD TRUE CHARACTERISTIC
        STO   1 RIGHT-1-R1PNT
        BSC   I NORM
NT2     DC      0
EXC     LDX   L2 R2PNT
        LD    2 K1-R2PNT
        MDX     ADDQ
*
*
*
*               SCHR
*               CHANGES SIGN OF RIGHT
*
*
SCHR    DC      0
        STS   2 STAT-R2PNT      RESET OVERFLOW
        SLT     32
        SD    1 RIGHT-R1PNT
        BSC   L SOVFL,0         BRANCH ON OVERFLOW
SPUT    STD   1 RIGHT-R1PNT
        BSC   I SCHR
SOVFL   LDD   2 H7FFF-R2PNT CAN ONLY OCCUR IF /8000 00
        MDX     SPUT            WAS IN AC * Q
*
*
*
*               FDIV
*               INPUT IN LEFT AND RIGHT
*               OUTPUT IN RIGHT
*
*
FDIV    DC      0
        STS   2 STAT-R2PNT
        LD    1 RIGHT-R1PNT
        BSC   L DIVCK,-4 B IF ZERO DIVISOR
        LD    1 LEFT-1-R1PNT
        S     1 RIGHT-1-R1PNT
        A     2 K129-R2PNT      RESTORE BIAS
        STO   1 RIGHT-1-R1PNT
        LD    1 RIGHT+1-R1PNT
        SRA     2
        M     1 LEFT-R1PNT
        D     1 RIGHT-R1PNT
        EOR   2 HFFFF-R2PNT
        A     2 K1-R2PNT
        SRT     14
        AD    1 LEFT-R1PNT
        SRT     2
        BSC     0               OVERFLOW CHECK
        EOR   2 HC000-R2PNT
        D     1 RIGHT-R1PNT
        STO   1 QUOTE-R1PNT
        SLT     16
        SRT     1
        D     1 RIGHT-R1PNT
        SRT     15
        A     1 QUOTE-R1PNT
        STD   1 RIGHT-R1PNT
        BSI   2 NORM-R2PNT
        BSI   2 FRNG-R2PNT
DIVF    BSC   I FDIV
```

```
*                                                                  CF013870
*                                                                  CF013880
*              DIVCK                                               CF013890
*              CHECKS FOR DIVISION BY ZERO                         CF013900
*                                                                  CF013910
*                                                                  CF013920
DIVCK LD     1 LEFT-R1PNT                                          CF013930
*                                                                  CF013940
*              SECOND ARGUMENT IS KNOWN TO BE ZERO                 CF013950
      BSC  L   EQ1,+-          BRANCH IF DIVIDEND ZEROCF013960
      BSI  L   ERROR                                              CF013970
      DC       3               ERROR 3 IS DIVISION BY CF013980
*              ZERO OF A NON ZERO QUANTITY                         CF013990
EQ1   LDD    2 ONE-R2PNT                                         CF014000
      BSI    3 FLDR-R3PNT                                        CF014010
      MDX      DIVF                                              CF014020
*                                                                  CF014030
*                                                                  CF014040
*                                                                  CF014050
*                                                                  CF014060
*                                                                  CF014070
*                                                                  CF014080
*            IFIX                                                  CF014090
*              CONVERTS FLOATING TO INTEGER                        CF014100
*      INPUT IN RIGHT                                              CF014110
*      OUTPUT IN RIGHT                                             CF014120
*      ALSO IN AC                                                  CF014130
*              ABSOLUTE VALUE OF INTEGER RETURNED                  CF014140
*                                                                  CF014150
*                                                                  CF014160
IFIX  DC       0                                                   CF014170
      BSI    1 FABS-R1PNT                                        CF014180
      BSI    2 SWAP-R2PNT      SWITCH ARGUMENTS TO LEFCF014190
      LDD    2 P5-R2PNT        %0.5                              CF014200
      BSI    3 FLDR-R3PNT      ADD  .5 FOR ROUNDING              CF014210
      BSI    2 FADD-R2PNT      FOR ROUNDING                      CF014220
      LD     1 RIGHT-1-R1PNT   BIASED EXPONENT                   CF014230
      S      2 K128-R2PNT      DELETE BIAS                       CF014240
      BSC  L   EOK,Z-          BRANCH IF EXPONENT -              CF014250
      LD     2 ZERO-R2PNT      STORE ZERO AS INTEGER             CF014260
      MDX      IRTN            RETURN                            CF014270
EOK   AND    2 H000F-R2PNT     LIMIT SHIFT RANGE                 CF014280
*                                                                  CF010770
      EOR    2 HFFFF-R2PNT     CHANGE TO NEGATIVE                CF014290
      A        ISHFT           ADD TO SHIFT INSTRUCTIOCF014300
      STO      *+1                                               CF014310
      LD     1 RIGHT-R1PNT                                       CF014320
      DC       *-*             SHIFT RIGHT                       CF014330
IRTN  STO    1 RIGHT-R1PNT                                       CF014340
      BSC  I   IFIX                                              CF014350
ISHFT SRT      16                                                CF014360
*                                                                  CF014370
*                                                                  CF014380
*                                                                  CF014390
*              LOADR + LOADL                                       CF014400
*              RESTORES SAVED RIGHT AND LEFT                       CF014410
*                                                                  CF014420
*                                                                  CF014430
LOADR DC       0                                                   CF014440
      LDD    1 RSAVE-R1PNT                                       CF014450
      STD    1 RIGHT-R1PNT                                       CF014460
      LD     1 RSAVE-1-R1PNT                                     CF014470
```

```
         STO   1 RIGHT-1-R1PNT                                  CF01448
         BSC   I LOADR                                          CF01449
*                                                               CF01450
*                                                               CF01451
*                                                               CF01452
LOADL    DC                                                     CF01453
         MDX   1 -4                                             CF01454
         BSI     LOADR                                          CF01455
         MDX   1 4                                              CF01456
         BSC   I LOADL                                          CF01457
*                                                               CF01458
*                                                               CF01459
*                                                               CF01460
*              FXPN                                             CF01461
*              TAKES EXP>B+                                     CF01462
*              B FOUND IN RIGHT                                 CF01463
*                                                               CF01464
*              ANSWER LEFT IN RIGHT                             CF01465
*                                                               CF01466
*                                                               CF01467
FXPN     DC      0                                              CF01468
         STS   2 STAT-R2PNT                                     CF01469
         LD    1 RIGHT-1-R1PNT                                  CF01470
         S     2 K143-R2PNT                                     CF01471
         BSC   L QBIG+-                                         CF01472
         LDD     LOG2E                                          CF01473
         BSI   2 XMDS-R2PNT                                     CF01474
         STD   1 RIGHT-R1PNT                                    CF01475
         LD    1 RIGHT-1-R1PNT                                  CF01476
         A     2 K1-R2PNT                                       CF01477
         STO   1 RIGHT-1-R1PNT                                  CF01478
         LD    2 K143-R2PNT                                     CF01479
         BSI   2 MOVER-R2PNT                                    CF01480
         A     2 K129-R2PNT                                     CF01481
         STO   1 QSAVE-R1PNT                                    CF01482
         SLA     16                                             CF01483
         SLT     2                                              CF01484
         SLA     1                                              CF01485
         A       M8CON                                          CF01486
         STO     D+1                                            CF01487
         STO     DZ+1                                           CF01488
         LD    2 K126-R2PNT                                     CF01489
         BSI   2 MOVER-R2PNT                                    CF01490
         AND   2 H7FFF-R2PNT                                    CF01491
         S     2 H4000-R2PNT                                    CF01492
         STD   1 RIGHT-R1PNT                                    CF01493
         LDD     A4EXP                                          CF01494
         BSI   2 XMDS-R2PNT                                     CF01495
         AD      A3EXP                                          CF01496
         BSI   2 XMDS-R2PNT                                     CF01497
         AD      A2EXP                                          CF01498
         BSI   2 XMDS-R2PNT                                     CF01499
         AD      A1EXP                                          CF01500
         BSI   2 XMDS-R2PNT                                     CF01501
         STD   1 RIGHT-R1PNT                                    CF01502
D        LDD   L *-*                                            CF01503
         BSI   2 XMDS-R2PNT                                     CF01504
DZ       AD    L *-*                                            CF01505
         STD   1 RIGHT-R1PNT                                    CF01506
         LD    1 QSAVE-R1PNT                                    CF01507
         STO   1 RIGHT-1-R1PNT                                  CF01508
```

```
         BSC  L  RNGER,0                                    CF01500
         BSI    2 FRNG-R2PNT                                CF01510
FNRTN    BSC  I  FXPN                                       CF01511
RNGER    BSI  L  ERROR                                      CF01512
         DC      1                                          CF01513
QBIG     LD     1 RIGHT-R1PNT                               CF01514
         BSC  L  RNGER,-                                    CF01515
         SLT     32                                         CF01516
         BSI    3 FLDR-R3PNT                                CF01517
         MDX     FNRTN                                      CF01518
M8TAB    DEC     1.09050773472525881                        CF01519
         DEC     1.29683955485101081                        CF01520
         DEC     1.54221082580794181                        CF01521
         DEC     1.83400808650934281                        CF01522
LOG2F    DEC     1.44269504128896381                        CF01523
A1EXP    DEC     0.693147079B2                              CF01524
A2EXP    DEC     0.240226486B4                              CF01525
A3EXP    DEC     0.055530155786                             CF01526
A4EXP    DEC     0.009621739988                             CF01527
M8CON    DC      M8TAB                                      CF01528
*                                                           CF01529
R3PNT    EQU     *                                          CF01530
*                                                           CF01531
*                                                           CF01532
*                CONSTANTS UNDER R3PNT                      CF01533
*                                                           CF01534
RSLT1    DEC     0                                          CF01535
PREC     DC      0                                          CF01536
NMDIG    DC      0                                          CF01537
POINT    DC      0                                          CF01538
NMOPL    DC      0                                          CF01539
C0       DC      /00F0                                      CF01540
LMASK    DC      /00BF                                      CF01541
LEVEL    DC      0                                          CF01542
H0030    DC      /0030                                      CF01543
K50      DC      50                                         CF01544
LINE     DC      0                                          CF01545
LITL     DC      /0093                                      CF01546
DECPT    DC      /004B                                      CF01547
LITO     DC      /0096                                      CF01548
MFLAG    DC      0              SIGN FLAG                   CF01549
LPRN     DC      /004D          LEFT PARENTHESIS            CF01549
AFLG     DC      0                                          CF01549
RFLG     DC      0                                          CF01549
NFLG     DC      0                                          CF01549
*                                                           CF01550
*                                                           CF01551
*                                                           CF01552
*                FLN                                        CF01553
*                TAKES NATURAL LOG OF B                     CF01554
*                WHERE B IS IN RIGHT                        CF01555
*                ANSWER IS RETURNED TO RIGHT                CF01556
*                                                           CF01557
*                                                           CF01558
FLN      DC      0                                          CF01559
         LD    1 RIGHT-1-R1PNT                              CF01560
         S     2 K128-R2PNT                                 CF01561
         STO   1 J-R1PNT                                    CF01562
         LD    2 K128-R2PNT                                 CF01563
         STO   1 RIGHT-1-R1PNT                              CF01564
         LDD   1 RIGHT-R1PNT                                CF01565
```

```
         BSC  L  FOK,Z-                                          CF015...
         BSI  L  ERROR                                           CF015...
         DC      21                    LN OF 0 OR -N             CF015...
FOK      SD      SQRH                                            CF015...
         BSC  L  RATIO,-                                         CF015...
         LD    2 K129-R2PNT                                      CF015...
         STO   1 RIGHT-1-R1PNT                                   CF015...
         LD    1 J-R1PNT                                         CF015...
         S     2 K1-R2PNT                                        CF015...
         STO   1 J-R1PNT                                         CF015...
RATIO    BSI   2 SWAP-R2PNT                                      CF015...
         BSI   2 SAVER-R2PNT                                     CF015...
         LDD   2 ONE-R2PNT                                       CF015...
         BSI   3 FLDR-R3PNT                                      CF015...
         BSI   2 FSUB-R2PNT                                      CF015...
         BSI   2 SWAP-R2PNT                                      CF015...
         LDD   2 TWO-R2PNT                                       CF015...
         BSI   3 FLDR-R3PNT                                      CF015...
         BSI   2 FADD-R2PNT                                      CF015...
         BSI   2 FDIV-R2PNT                                      CF015...
         BSI   2 SWAP-R2PNT                                      CF015...
         LD    2 K126-R2PNT                                      CF015...
         BSI   2 MOVEL-R2PNT                                     CF015...
         STD   1 RIGHT-R1PNT                                     CF015...
*                                                                CF015...
*                                                                CF015...
         BSI   2 XMDS-R2PNT                                      CF015...
         STD   1 RIGHT-R1PNT                                     CF015...
         SRT     10                                              CF015...
         AD      A4LN                                            CF015...
         BSI   2 XMDS-R2PNT                                      CF015...
         AD      A2LN                                            CF015...
         BSI   2 XMDS-R2PNT                                      CF015...
         SRT     6                                               CF015...
         A     2 A0LN-R2PNT                                      CF016...
         STD   1 RIGHT-R1PNT                                     CF016...
         LD    2 K130-R2PNT                                      CF016...
         STO   1 RIGHT-1-R1PNT                                   CF016...
         BSI   2 FMPY-R2PNT                                      CF016...
         BSI   2 SWAP-R2PNT                                      CF016...
*                                                                CF016...
*                                                                CF016...
         LD    1 J-R1PNT                                         CF016...
         SRT     16                                              CF016...
         BSI   1 FLOAT-R1PNT                                     CF016...
*                                                                CF016...
         BSI   2 SAVEL-R2PNT                                     CF016...
         LDD     LN2                                             CF016...
         STD   1 LEFT-R1PNT                                      CF016...
         LD    2 K128-R2PNT                                      CF016...
         STO   1 LEFT-1-R1PNT                                    CF016...
         BSI   2 FMPY-R2PNT                                      CF016...
         BSI   3 LOADL-R3PNT                                     CF016...
         BSI   2 FADD-R2PNT                                      CF016...
*                                                                CF016...
         BSI   2 SWAP-R2PNT                                      CF016...
         BSI   3 LOADR-R3PNT                                     CF016...
         BSI   2 SWAP-R2PNT                                      CF016...
         BSC   I FLN                                             CF016...
A0LN     EQU     H4000                                           CF016...
A2LN     DEC     0.6666441378680                                 CF016...
```

```
A4LN    DEC     0.401923469784                                  CF01627
SQRH    DEC     1.41421481                                      CF01628
LN2     DEC     0.69314718065994528O                            CF01629
*                                                               CF01630
*                                                               CF01631
*               FLDR + FLDL                                     CF01632
*               TAKE TWO WORD NUMBER FROM AC + O AND            CF01633
*               PUT THEM AS THREE WORD FLOATING                 CF01634
*               POINT IN RIGHT OR LEFT RESPECTIVFLY             CF01635
*                                                               CF01636
FLDR    DC      0                                               CF01637
        STO   1 RIGHT-R1PNT                                     CF01638
        RTE     8                                               CF01639
        SRA     8                                               CF01640
        STO   1 RIGHT-1-R1PNT                                   CF01641
        EOR   1 RIGHT+1-R1PNT                                   CF01642
        STO   1 RIGHT+1-R1PNT                                   CF01643
        BSC   I FLDR                                            CF01644
FLDL    DC      0                                               CF01645
        MDX   1 -4                                              CF01646
        BSI     FLDR                                            CF01647
        MDX   1 4                                               CF01648
        BSC   I FLDL                                            CF01649
*                                                               CF01650
*                                                               CF01651
*                                                               CF01652
        END     ERROR                                           CF01653
// DUP
*DUMP        WS  CD  CF000
```

```
/ JOB
/ ASM
_IST
PRINT SYMBOL TABLE
                *               FUNCTION FCALC                                  ICNV0010
                *                                                               ICNV0020
                *               FUNCTION READS IN 1 OR MORE                     ICNV0030
                *               SECTORS, DEPENDING ON NMSEK                     ICNV0040
                *               THEY ARE FOUND ON THE SYSTEM PACK AT            ICNV0050
                *               SECTOR DENOTED BY SECAD.                        ICNV0060
                *               THEY ARE LOADED INTO LOCATIONS                  ICNV0070
                *               STARTING AT OVLST.                              ICNV0080
                *               TRANSFER IS MADE TO OVLST +2 --                 ICNV0090
                *               AS A SUBROUTINE CALL.                           ICNV0100
                *               SUBROUTINE IS LOADED INTO CORE                  ICNV0110
                *               ONLY IF IT IS NOT ALREADY IN CORE.              ICNV0120
                *               CHECK IS MADE TO SEE IF IT IS THERE BY          ICNV0130
                *               CHECKING THE FIRST TWO WORDS                    ICNV0140
                *               OF THE PROGRAM AREA.                            ICNV0150
                *               IF THEY ARE THE RIGHT WORD COUNT                ICNV0160
                *               AND SECTOR ADDRESS RESPECTIVELY                 ICNV0170
                *               IT IS ASSUMED THAT THE SUBROUTINE               ICNV0180
                *               IS ALREADY IN CORE.                             ICNV0190
                *               UPON RETURN FROM SUBROUTINE,                    ICNV0200
                *               IF INPUT PARAMETER IS 3 OR 4, VALUE IN          ICNV0210
                *               RSLT1 IS CONVERTED TO CHARACTERS AND            ICNV0220
                *               PLACED IN BUFFER 0.  IF PARAMETER IS            ICNV0230
                *               NOT 3 OR 4, THE FUNCTION RETURNS TO             ICNV0240
                *               THE SYSTEM.                                     ICNV0250
                *                                                               ICNV0260
                OVLST   EQU     /6800           APPLICATION 2 AREA - 2          ICNV0270
                GGET    EQU     /00AE           DISK READ AREA                  ICNV0280
                DKERR   EQU     /00E5           DISK ERROR RETURN               ICNV0290
                NMSEK   EQU     6               NM SECTORS TO READ              ICNV0300
                SECAD   EQU     270             SECTOR NUM OF PROGRAM           ICNV0310
                SECTR   EQU     320             NUMBER OF WORDS PER SEC         ICNV0320
                *                               ON SYSTEM DISK                  ICNV0330
                ATR     EQU     /1D7C                                           ICNV0340
                ATRFR   EQU     ATR+374         FUNCTION RETURN                 ICNV0350
                *                                                               ICNV0360
                *                                                               ICNV0370
                        MDX     FSTRT                                           ICNV0380
                        DC      /FFFF                                           ICNV0390
                ACQSV   DEC     0               SAVE AC AND Q                   ICNV0400
                WDCNT   DC      SECTR*NMSEK     NM WDS TO READ                  ICNV0410
                MASK    DC      /0FFF                                           ICNV0420
                ADDR    DC      SECAD                                           ICNV0430
                FSTRT   STX  L1 XR1SV           SAVE REGISTERS                  ICNV0440
                        STX  L2 XR2SV                                           ICNV0450
                        STX  L3 XR3SV                                           ICNV0460
                        STD     ACQSV SAVE AC + Q                               ICNV0470
                        LD   L  OVLST           CHECK IF PROGRAM IN             ICNV0480
                *                               CORE                            ICNV0490
                        S       WDCNT                                           ICNV0500
                        BSC  L  LOAD,Z   BRANCH IF NOT SAME                     ICNV0510
                        LD   L  OVLST+1         CHECK SECTOR ADDRESS            ICNV0520
                        AND     MASK            MASK DRIVE NUMBER               ICNV0530
                        S       ADDR            SECTOR ADDRESS                  ICNV0540
                        BSC     +-              BRANCH IF NOT SAME              ICNV0550
                        MDX     BRNCH                                           ICNV0560
                LOAD    LDX  L1 SECTR*NMSEK     LOAD PROGRAM                    ICNV0570
```

```
         STX   L1 OVLST                NUMWDS IN OVLST                ICNV0580
         LDX   L1 SECAD                SECTOR ADDRESS                 ICNV0590
         STX   L1 OVLST+1              IN OVLST+1                     ICNV0600
         LDX   L1 OVLST                XR1 POINTS TO OVLST            ICNV0610
         SLT      32                   AC CONTAINS  LOGICAL 0         ICNV0620
         BSI   I  GGET                 GET PROGRAM                    ICNV0630
         DC       DKERR                ERROR RETURN                   ICNV0640
BRNCH    BSI   L  OVLST+2              BRANCH TO PROGRAM              ICNV0650
HFF00    EQU      /6D21          2                                    ICNV0660
HFFFF    EQU      /6D22          2                                    ICNV0670
FXPN     EQU      /6E17          3                                    ICNV0680
FSUB     EQU      /6D24          2                                    ICNV0690
FLOG     EQU      /6CF1          1                                    ICNV0700
FLDR     EQU      /6EC2          3                                    ICNV0710
FLDL     EQU      /6ECB          3                                    ICNV0720
FDIV     EQU      /6DCB          2                                    ICNV0730
FAXB     EQU      /6C62          1                                    ICNV0740
DECPT    EQU      /6E73          3                                    ICNV0750
C0       EQU      /6E6C          3                                    ICNV0760
R1PNT    EQU      /6CCC          1                                    ICNV0770
R2PNT    EQU      /6D7B          2                                    ICNV0780
R3PNT    EQU      /6E65          3                                    ICNV0790
PADR     EQU      /69A0          -                                    ICNV0800
MADR     EQU      /699F          -                                    ICNV0810
SCHR     EQU      /6DC0          2                                    ICNV0820
SAVEL    EQU      /6D48          2                                    ICNV0830
RSLT1    EQU      /6E66          3                                    ICNV0840
RIGHT    EQU      /6CE2          1                                    ICNV0850
SWAP     EQU      /6D6C          2                                    ICNV0860
QUOTE    EQU      /6CDA          1                                    ICNV0870
PACKR    EQU      /6CB2          1                                    ICNV0880
ONE      EQU      /6D04          2                                    ICNV0890
NMOPL    EQU      /6E6B          3                                    ICNV0900
MFLAG    EQU      /6E75          3                                    ICNV0910
LOADL    EQU      /6E11          3                                    ICNV0920
LINE     EQU      /6E71          3                                    ICNV0930
LEVEL    EQU      /6E6E          3                                    ICNV0940
K5       EQU      /6D14          2                                    ICNV0950
K128     EQU      /6D18          2                                    ICNV0960
K2       EQU      /6D11          2                                    ICNV0970
K1       EQU      /6D10          2                                    ICNV0980
IFIX     EQU      /6DF4          3                                    ICNV0990
H000F    EQU      /6D1D          2                                    ICNV1000
H00FF    EQU      /6D1E          2                                    ICNV1010
ATRE6    EQU      ATR+370              ADDR OF FN PARAMETER           ICNV1020
ATRB0    EQU      ATR+35               ADDR OF BUFFER ZERO            ICNV1030
SIGNF    EQU      FXPN           3                                    ICNV1040
SIGNP    EQU      RIGHT-2        1                                    ICNV1050
LEFTC    EQU      FAXB           1                                    ICNV1060
WHOLF    EQU      MFLAG          3                                    ICNV1070
FRAC     EQU      LINE           3                                    ICNV1080
PWR      EQU      LEVEL          3                                    ICNV1090
PWRX     EQU      NMOPL          3                                    ICNV1100
RSLT2    EQU      QUOTE          1                                    ICNV1110
F0       EQU      C0             3                                    ICNV1120
SAVE     EQU      NMOPL          3                                    ICNV1130
         LD    I  ATRE6                PICK UP FN PARAMETER           ICNV1140
         SLA      4                                                   ICNV1150
         SRA      12                                                  ICNV1160
         S     2  K2-R2PNT             CHECK FOR GT 2                 ICNV1170
         BSC   L  FRTN+4               BRANCH IF NO RESULT            ICNV1180
```

```
*                                          TO BE RETURNED                ICNV1190
                                                                         ICNV1200
        LD      BOAD
        STO   L CURAD                      RESET PTNXT                   ICNV1210
        SLA     15                         RESET                         ICNV1220
        STO     3 SIGNF-R3PNT              SIGN OF FRACTION              ICNV1230
        STO     1 SIGNP-R1PNT              SIGN OF POWER                 ICNV1240
        STO     1 LEFTC-R1PNT              LEFT CHARACTER FLAG           ICNV1250
        LDD     3 RSLT1-R3PNT                                            ICNV1260
        BSI     3 FLDR-R3PNT               PUT IN RIGHT                  ICNV1270
        LD      3 RSLT1-R3PNT              TEST FOR MINUS                ICNV1280
        BSC   L C0000,-                    BRANCH IF NOT                 ICNV1290
        STO     3 SIGNF-R3PNT              SIGN OF FRACTION              ICNV1300
        BSI     2 SCHR-R2PNT                                             ICNV1310
        BSI     1 PACKR-R1PNT                                            ICNV1320
        STD     3 RSLT1-R3PNT                                            ICNV1330
        BSI     3 FLDR-R3PNT                                             ICNV1340
        LDD     3 RSLT1-R3PNT                                            ICNV1350
C0000 BSC   L C0001,Z                                                    ICNV1360
        STO     3 WHOLE-R3PNT              STORE ZERO IN WHOLE           ICNV1370
        STO     3 PWR-R3PNT                AND POWER                     ICNV1380
        STO     1 RSLT2-R1PNT              AND FRACTION SUM              ICNV1390
        BSC   L PUTIT                      PUT IN BUFFER                 ICNV1400
BOAD    DC      ATRB0                      ADDRESS OF BUFFER 0           ICNV1410
C0001 BSI     1 FLOG-R1PNT                 LOG OF RESULT                 ICNV1420
        BSI     1 PACKR-R1PNT              IN AC + 0                     ICNV1430
        BSC     Z+                                                       ICNV1440
        STO     1 SIGNP-R1PNT                                            ICNV1450
        BSI     3 FLDR-R3PNT                                             ICNV1460
        BSI     3 IFIX-R3PNT                                             ICNV1470
C0004 STO     3 PWR-R3PNT                  INTEGER ABSOLUTE VALUE        ICNV1480
        STO     3 PWRX-R3PNT               OF LOG OF RESULT              ICNV1490
        LD      1 SIGNP-R1PNT              SIGN OF POWER                 ICNV1500
        BSC     -                                                        ICNV1510
        MDX     C0002                      POSITIVE POWER                ICNV1520
        LD    L MADR                       - POWER                       ICNV1530
        MDX     *+2                                                      ICNV1540
C0002 LD    L PADR                         Z+ POWER                      ICNV1550
        STO     TABPT                                                    ICNV1560
        LDD     3 RSLT1-R3PNT              PUT RESULT                    ICNV1570
        BSI     3 FLDL-R3PNT               IN LEFT                       ICNV1580
        LD      3 PWRX-R3PNT               LOAD POWER OF TEN             ICNV1590
PTEST BSC   L DIVOV,+-                     BRANCH IF ZERO                ICNV1600
        BSC     E                                                        ICNV1610
        MDX     *+1                                                      ICNV1620
        MDX     LBOFF                      BRANCH IF LOW BIT OFF         ICNV1630
TABPT   EQU     *+1                                                      ICNV1640
        LDD   L *-*                        LOAD POWER OF TEN             ICNV1650
        BSI     3 FLDR-R3PNT               PUT IN RIGHT                  ICNV1660
        BSI     2 FDIV-R2PNT               DIVIDE RESULT                 ICNV1670
        BSI     2 SWAP-R2PNT               PUT IN LEFT                   ICNV1680
LBOFF MDX   L TABPT,2                      BUMP POWER POINTER            ICNV1690
        LD      3 PWRX-R3PNT                                             ICNV1700
        SRA     1                                                        ICNV1710
        STO     3 PWRX-R3PNT                                             ICNV1720
        MDX     PTEST                                                    ICNV1730
DIVOV BSI     2 SAVFL-R2PNT                SAVE DIVIDED RESULT           ICNV1740
        BSI     2 SWAP-R2PNT               PUT IN RIGHT                  ICNV1750
        BSI     1 PACKR-R1PNT              RESULT IN AC + 0              ICNV1760
        STD     1 RSLT2-R1PNT              RETURN TO TESLT2              ICNV1770
        BSI     3 LOADL-R3PNT              PUT IN LEFT                   ICNV1780
        LD      1 SIGNP-R1PNT              CHECK POWER SIGN              ICNV1790
```

```
        BSC  L  C0014,-          BRANCH IF GE ZERO          ICNV1800
        LDD  I  PADR             %10.0                      ICNV1810
        BSI     3 FLDR-R3PNT     SEE IF                     ICNV1820
        BSI     2 FSUB-R2PNT     DIVIDED RESULT IS          ICNV1830
        LD      1 RIGHT-R1PNT    GE 10.0                    ICNV1840
        BSC  L  C0020,Z+                                    ICNV1850
        MDX     C0015            BRANCH IF SO               ICNV1860
C0014   LDD     2 ONE-R2PNT      %1.0                       ICNV1870
        BSI     3 FLDR-R3PNT     SEE IF                     ICNV1880
        BSI     2 FSUB-R2PNT     DIVIDED RESULT IS          ICNV1890
        LD      1 RIGHT-R1PNT    LT 1.0                     ICNV1900
        BSC  L  C0003,-          BRANCH IF NOT              ICNV1910
C0015   LD      3 PWR-R3PNT      DECREMENT POWER            ICNV1920
        S       2 K1-R2PNT       AND REPEAT                 ICNV1930
        MDX     C0004            DIVISION                   ICNV1940
C0020   LDD     2 ONE-R2PNT                                 ICNV1950
        BSI     3 FLDR-R3PNT                                ICNV1960
        BSI     2 FSUB-R2PNT                                ICNV1970
        LD      1 RIGHT-R1PNT                               ICNV1980
        BSC  L  C0003,-                                     ICNV1990
        LD      3 PWR-R3PNT                                 ICNV2000
        A       2 K1-R2PNT                                  ICNV2010
        MDX     C0004                                       ICNV2020
ISHFT   SRT     16                                          ICNV2030
C0003   LDD     1 RSLT2-R1PNT    DIVIDED RESULT ≥DR↑        ICNV2040
        BSI     3 FLDR-R3PNT     IN RIGHT                   ICNV2050
        LD      1 RIGHT-1-R1PNT  EXPONENT OF DR             ICNV2060
        S       2 K128-R2PNT     SUBTRACT /80               ICNV2070
        BSC  L  EOK,Z-                                      ICNV2080
        SLT     32               SHOULD NEVER BE            ICNV2090
        MDX     C0005            EXECUTED                   ICNV2100
EOK     AND     2 H000F-R2PNT    LIMIT SHIFT RANGE          ICNV2110
        EOR     2 HFFFF-R2PNT    COMPLEMENT                 ICNV2120
        A       ISHFT                                       ICNV2130
        STO     *+1                                         ICNV2140
        LDD     1 RIGHT-R1PNT    PICK UP DR                 ICNV2150
        DC      *-*              SHIFT                      ICNV2160
C0005   STO     3 WHOLE-R3PNT                               ICNV2170
        SLT     16                                          ICNV2180
        STO     3 FRAC-R3PNT                                ICNV2190
        LD      SADDR            ADDRESS OF STABLE          ICNV2200
        STO     SPNTR            INTO SUM ROUTINE           ICNV2210
        SLT     32               ZERO OUT RSLT2 TO          ICNV2220
        STD     1 RSLT2-R1PNT    CONTAIN SUM                ICNV2230
        LD      3 FRAC-R3PNT                                ICNV2240
STEST   BSC  L  ADDOV,+-                                    ICNV2250
        BSC  L  HBOFF,-          BRANCH ON HIGH BIT OFF     ICNV2260
SPNTR   EQU     *+1              IF BIT IS ON               ICNV2270
        LDD  L  *-*              ADD IN TO SUM THE          ICNV2280
        AD      1 RSLT2-R1PNT    CORRESPONDING              ICNV2290
        STD     1 RSLT2-R1PNT    FRACTIONAL AMOUNT          ICNV2300
HBOFF   MDX  L  SPNTR,2          BUMP FRAC POINTER          ICNV2310
        LD      3 FRAC-R3PNT     SHIFT FRAC                 ICNV2320
        SLA     1                1 PLACE                    ICNV2330
        STO     3 FRAC-R3PNT     TO LEFT                    ICNV2340
        MDX     STEST            TEST AGAIN                 ICNV2350
SADDR   DC      STABL            ADDRESS OF S TABLE         ICNV2360
ADDOV   LDD     1 RSLT2-R1PNT    FINAL SUM                  ICNV2370
        D       TNTH             %10000                     ICNV2380
        A       2 K5-R2PNT       ROUNT FACTOR               ICNV2390
        SRT     16               PUT IN Q                   ICNV2400
```

```
        D       K10             ROUNDED FRAC SUM
        STO   1 RSLT2-R1PNT     IN RSLT2
        S       K1000
        BSC   L C0021,Z+        BRANCH IF LT 1000
        STO   1 RSLT2-R1PNT
        LD    3 WHOLE-R3PNT     BUMP WHOLE NUMBER
        A     2 K1-R2PNT
        STO   3 WHOLE-R3PNT
C0021   LD    3 WHOLE-R3PNT
        S       K10
        BSC   L C0006,Z+        BRANCH IF LT 10
        LD    2 K1-R2PNT        SET WHOLE TO 1
        STO   3 WHOLE-R3PNT     AND
        LD    1 SIGNP-R1PNT     CHECK POWRER SIGN --
        BSC   L C012,+-         IF NEGATIVE
        LD    3 PWR-R3PNT       DECREMENT
        S     2 K1-R2PNT        POWER
        MDX     C013            BY 1
C012    LD    2 K1-R2PNT        IF POSITIVE
        A     3 PWR-R3PNT       BUMP POWER BY 1
C013    EQU     *               STORE IN PWR
        STO   3 PWR-R3PNT
C0006   EQU     *
PUTIT   LD    3 SIGNF-R3PNT
        BSC   L C0007,+-        BRANCH IF SIGN OFF
        LD      MINUS           PUT - IN BUFFER
        BSI     PTNXT
C0007   LD    3 WHOLE-R3PNT     PUT INTEGER PORTION
        OR    3 F0-R3PNT        IN BUFFER
        BSI     PTNXT
        LD    3 DECPT-R3PNT     PUT DEC PT IN BUFFER
        BSI     PTNXT
        LD      K100            PUT HUNDREDS DIGIT OF
        BSI     PUTIN           FRACTION IN BUFFER
        LD      K10             PUT TENS DIGIT OF
        BSI     PUTIN           FRACTION IN BUFFER
        LD    2 K1-R2PNT        PUT ONES DIGIT OF
        BSI     PUTIN           FRACTION IN BUFFER
        LD    3 PWR-R3PNT       TEST POWER FOR ZERO
        BSC   L C0009,+-        BRANCH ON ZERO
        LD      BLANK           PUT BLANK IN BUFFER
        BSI     PTNXT
        LD      TIMES           PUT X IN BUFFER
        BSI     PTNXT
        LD      BLANK           PUT BLANK IN BUFFER
        BSI     PTNXT
        LD      F1              PUT 1 IN BUFFER
        BSI     PTNXT
        LD    3 F0-R3PNT        PUT 0 IN BUFFER
        BSI     PTNXT
        LD      RLF             PUT REVERSE LINE
        BSI     PTNXT           FEED IN BUFFER
        LD    1 SIGNP-R1PNT     TEST SIGN OF POWER
        BSC   L C0008,-         BRANCH ON PLUS
        LD      MINUS           PUT - IN BUFFER
        BSI     PTNXT
C0008   LD    3 PWR-R3PNT       PUT POWER IN PLACE
        STO   1 RSLT2-R1PNT     FOR PUTIN
        S       K10             TEST FOR LEADING ZERO
        BSC     Z+              SKIP IF NONE
        MDX     C0011           YES--DELETE IT
```

```
         LD       K10              PUT TEN DIGIT OF
         BSI      PUTIN            POWER IN BUFFER
C0011    EQU      *
         LD     2 K1-R2PNT         PUT ONES DIGIT OF
         BSI      PUTIN            POWER IN BUFFER
C0009    LD       EOB
         BSI      PTNXT
         MDX      FRTN
*
*                 PUTIN
*                 STORES NUMERIC DIGIT IN BUFFER
*
PUTIN    DC       0
         STO    1 RSLT2+1-R1PNT    SAVE DIVISOR
         LD     1 RSLT2-R1PNT      DIVIDE
         SRT      16
         D      1 RSLT2+1-R1PNT
         RTE      16               PUT REMAINDER BACK
         STO    1 RSLT2-R1PNT      IN RSLT2
         RTE      16               PUT QUOTIENT IN
         OR     3 F0-R3PNT         BUFFER
         BSI      PTNXT
         BSC  I   PUTIN
*
*                 PTNXT
*                 PUTS CHARACTER IN BUFFER
*
PTNXT    DC       0
         STO    3 SAVE-R3PNT       SAVE CHARACTER
CURAD    EQU      *+1
         LD   L   *-*              GET WORD FROM BUFFER
         RTE      16               PUT IN Q
         LD     1 LEFTC-R1PNT
         BSC  L   P0001,Z          BRANCH ON RIGHT
         LD     3 SAVE-R3PNT
         SLA      8
         STO    3 SAVE-R3PNT
         RTE      16               GET WORD FROM Q
         AND    2 H00FF-R2PNT      ZAP LEFT 8 BITS
P0000    OR     3 SAVE-R3PNT       OR IN CHARACTER
         STO  I   CURAD            RETURN WORD
         MDX      P0002
P0001    RTE      16               GET WORD FROM Q
         AND    2 HFF00-R2PNT      ZAP RIGHT 8 BITS
         MDX      P0000
P0002    LD     1 LEFTC-R1PNT      REVERSE LEFTC
         EOR    2 K1-R2PNT
         STO    1 LEFTC-R1PNT
         BSC  L   NOBMP,Z
         MDX  L   CURAD,1           BUMP WORD COUNTER
NOBMP    BSC  I   PTNXT
*
*                 CONSTANTS
*
K10      DC       10
K100     DC       100
K1000    DC       1000
BLANK    DC       /0040
EOB      DC       /0026
F1       DC       /00F1
MINUS    DC       /0060
```

```
RLF     DC      /0009                                          ICNV3630
TIMES   DC      /0081                                          ICNV3640
TNTH    DC      10000                                          ICNV3650
*                                                              ICNV3660
*       .       SUM TABLE                                      ICNV3670
*                                                              ICNV3680
STABL   DEC     50000000                                       ICNV3690
        DEC     25000000                                       ICNV3700
        DEC     12500000                                       ICNV3710
        DEC     06250000                                       ICNV3720
        DEC     03125000                                       ICNV3730
        DEC     01562500                                       ICNV3740
        DEC     00781250                                       ICNV3750
        DEC     00390625                                       ICNV3760
        DEC     00195313                                       ICNV3770
        DEC     00097656                                       ICNV3780
        DEC     00048828                                       ICNV3790
        DEC     00024414.                                      ICNV3800
        DEC     00012207                                       ICNV3810
        DEC     00006104                                       ICNV3820
        DEC     00003052                                       ICNV3830
        DEC     00001526                                       ICNV3840
        DEC     00000763                                       ICNV3850
FRTN    EQU     *                                              ICNV3860
        LDD  L  ACQSV           RESTORE AC + Q                 ICNV3870
XR1SV   EQU     *+1                                            ICNV3880
        LDX  L1 *-*             RESTORE REGISTERS              ICNV3890
XR2SV   EQU     *+1                                            ICNV3900
        LDX  L2 *-*                                            ICNV3910
XR3SV   EQU     *+1                                            ICNV3920
        LDX  L3 *-*                                            ICNV3930
        BSC  I  ATRFR           RETURN                         ICNV3940
        END     FSTRT                                          ICNV3950
' DUP                                                          ICNV3960
)UMP    WS  CD  FCALC
```